

16 bit isimler

AX	Accumulator
CX	Count
DX	Data
SP	Stack Pointer
BP	Base Pointer
DI	Destination Index
SI	Source Index
IP	Instruction Pointer

32 bit

16 bit

Flags

Genel Amaçlı Saklayıcılar

İşaretleme ve İndis Saklayıcıları

CS Code Segment

DS Data Segment

SS Stack Segment

ES Extra Segment

FS Özel isimli

GS Özel isimli

MOV ah, 0h

int 21h

END start

Turbo C++ IDE

File Edit Search Run Compile Debug Project Options Window

1998

MAKRO.C

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

int a,b;

BASLA

YAZ ("A sayısı:"); OKU ("%d",&a);

YAZ ("B sayısı:"); OKU ("%d",&b);

EGER (a>b) İSE YAZ ("A Büyük");

DEĞİLSE EĞER (a<b) İSE YAZ ("B Büyük");

DEĞİLSE YAZ ("Eşitler");

DUR

1:1

F1 Help F2 Save F3 Open Alt-F9 Compile F8 Make F10 Menu

UYGULAMALI GENEL PROGRAMLAMA

TARİK BAĞRIYANIK

Uygulamalı Genel Programlama

Bilgi paylaştıkça...

Gecenin derinliğinde
Dinleyen bilir, hisşırtıların
Ne kadar dinlendirici olduğunu

Tarık Bağrıyanık

İçindekiler

Giriş ve Kitabın Kullanımı	4
Yazar Hakkında	7
Yazılım Geliştirme ve Böcek Ayıklama	8
Bölüm 1 – Assembler	10
1. Mavi.txt (mavi renkli ekran)	14
2. meraba.asm (merhaba dünya!)	18
3. kucuk.asm (büyük harfe çevirir)	20
4. cizim.asm (klavye ile çizim yapma)	21
5. Fridge (intro yapımı)	23
Bölüm 2 – Basic	26
1. piyano.bas (müzik)	26
2. kutu.bas (kutu şekli)	27
Bölüm 3 – C Dili	29
1. sıralama (balon, seçerek ve hızlı sıralama)	35
2. makro.c (#define kullanımı)	38
3. unionreg.c (union tanımlama)	39
4. mousetst.c (fare kullanımı)	40
5. klvy.c (klavye ile çizim yapma)	42
6. domgunu.c (tarihin gününü bulma)	44
7. saat.c (DOS saati)	46
8. Grafikli.c (fareli grafik ekran)	48
Bölüm 4 – Pascal	54
1. txtall.pas (metin dosyaları açma)	55
2. diyet1.pas (grafik ekrana şekil çizme)	56
3. hesapmak.pas (hesap makinesi yapımı)	58
4. matrice.pas (matrisler)	60
Bölüm 5 – Delphi	62
1. Agent (Microsoft Agent kullanımı)	69
2. CD (görev çubuğunda program çalıştırma)	76
3. create (nesne oluşturma)	81
4. Dergilerim (basit veritabanı uygulaması)	84
5. DLL (Dynamic Linking Library)	89
6. Meraba (resimli ekran koruyucusu)	93
7. ParalelPort (Win9x için paralel veriyolu kullanımı)	103
8. Saat (yuvarlak form ve çizim yapılması)	111
9. Sozluk (İngilizce Türkçe sözlük)	115
10. surukle (nesne sürükleme ve yazıcı kullanımı)	117
11. Thread (işlemciyi daha verimli kullanma)	120
12. Sysmenu (sistem menüsüne eleman eklenmesi)	126
13. Tuş Basımı (klavye ve fare basımlarını kontrol etme)	127
14. 2 Boyutlu Oyun Yapımı (TicTacToe, Pazl, Mayn oyunları)	129
15. Gezgen (Windows Gezgini yapımı)	146
16. Dosya işlemi (Write ve Read komutları ile dosyalama)	152
17. Dosya Listele (dosya bulma işlemleri)	155
18. think2 (kendiliğinden dizilen nesneler)	160
19. DirectX ve OpenGL programcılığı (grafik programcılığı)	164
20. MS Peint (PaintBrush yapımı)	168
21. Sayı Bulmaca Oyunu (4 haneli sayı)	174
22. Veritabanı Grafik Analizi (DBChart kullanımı)	176
23. Okul Ders Programı (Haftalık ders programı yapımı)	179
24. Dolphin (müzik yapma programı)	191
25. Ofis (basit bir veritabanı uygulaması)	200
26. Tercuman (İngilizce'den Türkçe'ye çeviri ve okuma programı)	202
27. CGI kullanımı(Common Gateway Interface)	208
28. Öğrenci Bilgi İşleri (paradox veritabanı örnekleri)	211

29. Bit Filtrele (8 bit bilgiden seçim yapma)	213
30. Ekran Çizimi (ekrana çizim yaparak, ders anlatımı).....	215
31. Asansör (4 kat binanın asansör simülasyonu)	216
Bölüm 6 – Visual Basic	219
1. Sendkeys (diğer uygulamalara tuş basımı yollama)	219
2. Final (hesap makinesi yapımı)	220
3. Listbox (Print ve Input komutları ile dosyalama)	225
4. Fractal (pset grafik komutu)	228
Bölüm 7 – Diğer Diller ve çalışmalarım	231
1. Delphi ActiveX (web sayfasında ActiveX)	231
2. Excel yazıyla.xls (rakamı yazıya çevirme).....	234
3. Excel dosyalistesi.xls (dosya arama).....	236
4. Excel userform.xls (kullanıcı tanımlı formlar)	238
5. Excel ile tarih dökümü (modül kullanımı).....	240
6. WSH ile sayı toplama (*.vbs dosyalar)	242
7. WSH ile dosya listeleme (*.vbs ile Excel otomasyonu)	244
8. kopyala komutu (batch ve Pascal dili kullanarak)	245
9. JavaScript ile dosyalama (rasgele yazı kaydırılması)	249
10. JavaScript ve Flash (rasgele yazı kaydırılması)	254
11. JavaScript ile arama (*.html içinde arama).....	258
12. ASP ile misafir defteri (*.asp ve *.mdb veritabanı)	259
13. Yunus Programlama Dili (Türkçe açık kaynak programlama projesi).....	272
14. KüPro (Web tabanlı uygulama geliştirme).....	281
15. MyBoard (Flash ile dijital devre tasarımı).....	284
16. SiPro (Web tabanlı sınav programı geliştirme).....	287
17. Denklemler (Excel ile denklem çözümü)	296
18. ASP.NET uygulaması (Online Telefon Defteri).....	299
19. okul3d (Sanal Gerçeklik Örneği - VRML)	314
20. Net Pazar (PHP ve MySQL ile internette bit pazarı kurmak)	316
21. Trafik Simülasyonu (Flash Action Script)	319
22. Labirent Robotu Simülasyonu (Flash 8 Action Script)	322
23. dbase uygulamaları	325
24. Ev Kon (internet üzerinden ev kontrolü)	328
25. PIC (Peripheral Interface Controller)	334
Ek 1 – Algoritma Oluşturmak.....	341
Ek 2 – SQL	342
Ek 3 – ASCII Kod Tablosu	343
Ek 4 – InstallShield’in Kullanımı	345
Ek 5 – CD’de Neler Var?	349
Ek 6 – Sözlük	353
Dizin	357
Son söz.....	366
Kaynaklar	367

Giriş ve Kitabın Kullanımı

Kitap ve CD’deki örnekler, programcılığa heveslenen insanlarımıza basit, derine girmeden, öz ve faydalandırıcı şekilde vermeye çalışıldı. Ders kitabı olarak değil de, **örnekler** üzerine kurulu, pratik yapma, fikir verme, yöntemler gösterme ve kendi programcılık felsefenizi oluşturmaya yardımcı olmak amacı ile hazırladım. Aklınızda planladığınız ama nasıl yapıldığını kestiremediğiniz programlama örneklerinin benzerlerini burada bulabilirsiniz. Kitaptaki örnekleri CD’den bağımsız olarak da yapabilirsiniz. Bir türlü yaptığınız program çalışmıyorsa, yazım hatalarına ve kitapta verilen tüm kodlara dikkatle bakınız. Her ne kadar hepsi çalışıyor desem de donanım, işletim sistemi ve yazılım altyapısı farklı sistemlerde sorunlar çıkabiliyor. Mesela NT tabanlı bilgisayarlarda Paralel Port örneği bir işe yaramıyor, Delphi 6 ile yazdığım bazı programlar eski sürümlerde çalışmıyor veya bazı GWBasic programları Qbasic ile açamıyor gibi problemler var.

Elimden geldiğince **Türkçe**’mize de dikkat ederek kitabın açık olmasını sağlamaya gayret ettim. Metin kelimesi Türkçe kaynaklı mı bilmiyorum, fakat string ve text kelimeleri yerine metin kullandım. Bazı İngilizce kelimelerin de okunuşlarını italik olarak belirttim, kitap sonundaki sözlük ile takıldığınız kelimeleri bulabilirsiniz. Yine de siz yakınınızda bir sözlük bulundurun. Kod ve normal yazıyı farklı font – yazı tipi ile belirgin hale getirmeye çalıştım. Kod yazarken tek tırnak, çift tırnak ve boşlukların rahat okunması için Arial kullandım. Açıklama satırlarını bu fontun kalın olmayanı şeklinde hazırladım. Bazı durumlarda ` tek tırnak, " çift tırnak, 0 sıfır, o harfi, I büyük ı, l küçük le birbirine benziyor. Tuş birleşimleri "Alt+Tab" deyimindeki gibi + ile birleştirilerek yazdım. Menü komutları "" içinde araları * ile ayrılarak "Başlat*Çalıştır...*msconfig" gibi kullanılıyor. Bazı kodların **mavi** renk olması onlara dikkat edilmesi gerektiği anlamında kullandım. Normal yazımda ise kenarları düzgün olan Verdana yazı tipini seçtim. Bazı bölümlerde yer tasarrufu için iki kolon şeklinde böldüm. Fakat ilk kez okumada insanın "bunun devamı nerede?" diye kafası karışıyor. İlk kolonun devamı o sayfadaki ikinci kolondur, sonra diğer sayfanın ilk kolonuna devam edilir.

Fakat hala program kodlarında birçok hata ve açıklanması gerekli konular var. Size hiçbir yazılımın veya ürünün kusursuz olmadığını belirtmek istiyorum. Bu sebeple devamlı yeni sürüm programlar yazılıyor. Kitap içine yayılmış birçok **ipuçları** bulunuyor. Tek sorun bu ipuçlarına hemen ulaşamamanız. Çoğu örnekte, bu örnekte şu olayı da siz yapabilirsiniz, gibi deyimlerle çalışmanız için fikirler vermeye çalıştım. Yanlış anlamayın, tüm örnekler çalışır olduğu halde muhakkak siz de keşke şu özellik de olsaydı demişsinizdir.

İnsanımızın kitap okuma alışkanlığı olmadığı düşünülürse, örneklerle dayalı pratik yapmak daha ilgi çekici ve akılda kalıcı olduğuna inanıyorum. Bol bol örneklerle her an dilleri birbiri içine girmiş şekilde kitaba aktardım. Her bölümü adım adım geçmenize gerek yok. Özellikle kendi örneklerimin verirken gelişmeye açık olmalarına gayret ettim. Tabii benden daha yetenekli olan programcılarımızın çok daha ileriye gideceklerinden eminim. Belki de kitabı **dil çöplüğü** olarak da görebilirsiniz. Çoğu zaman öyle hissediyorum. Yani **bir** dili iyi bilmek de yeterli olur...

Belki de kafanızı en çok karıştıran anlatım stili, C dilinde bir örnek anlatırken bu işlevi Pascal’daki şu örnekte de yapsak gibi, kod dünyaları ayrı dillerde aynı beklentiye sahip olmak. Sizce de bu kod dönüşümleri olamaz mı? Örneğin VBA makrolarındaki tam aradığınız şeyi Delphi’de de yapamaz mısınız? C dilinde header, Pascal’da unit’ler, Delphi’de DLL ve ActiveX ile anlatmak istediğim ana düşünce, **global** kod oluşturmaktır. Ticari amaçla üretilen Microsoft Windows, açık kod sistemi ile geliştirilen C diline dayalı Unix de bu mantığa dayalı olarak devamlı ilerliyor. Microsoft firmasının hemen her yıl çıkardığı versiyonlu Windows, Office, VB ve DirectX kendi versiyonları arasında bile uyumsuz.

Yaklaşık 15 farklı dilden birçok örneği açıklamaları ile karşılaşırken bütün bu dillerin bilmek zorunda değilsiniz. Çok dil bilmekten daha önemli olan şey elinizdeki imkanları en iyi şekilde değerlendirmektir. Örnekleri noktası virgüle komut olarak açıklayamayacağım. Yani bilmediğimiz, ezberlemediğimiz birçok nokta olacak. Zaten buradaki amaç size tam olarak bu dilleri öğretmek değil, fikir vermek. Ne ile ne yapabilirsiniz gibi. Dile göre konu belirlemek yerine, konuya göre istediğimiz dilde program yazabilmek asıl hedefimiz. Ama genel olarak sitenize misafir defteri yerleştirmek istediğinizi düşünürsek, bunu ASP veya PHP dilleri ile yapabiliriz, hesap tabloları ile ilgilenmek istiyorsak Excel, Access bize uygun, ilgi alanımız donanım ise Assembler, C ve API’ler yeterlidir. Dünyaya sadece belli açılardan bakarak değil hedefimize göre uygun alet edevat seçerek ilerlemek gerekli. Artık yazılımlarda internet ağırlık kazanıyor. İnternet destekli olmayan programlar yavaş yavaş eleniyor. Temel neden güncellenebilirlik ve iletişim. İnternet, bilgisayar dergileri, bilgisayar kitapları ve okul bizim bilgi kaynaklarımız. delphiturk.com, programmersheaven.com, ve benzeri sitelerden her türlü kod ve programlama bilgisine, en güncel haberlere ve ipuçlarına dergileri takip ederek ulaşabilirsiniz.

Uğraşacağımız diller Microsoft Windows üzerine olacak. Donanım olarak ilk kullandığım intel386’dan beri hiçbir Pentium veya başka işlemcide sorun çıktığını görmedim. İlle de şu işletim sistemi ve işlemci olacak diye bir sorunumuz yok. Özellikle notepad - not defteri en büyük yardımcımız. "Windows\SendTo" içine not defterine kısayolu

oluşturarak, herhangi bir dosyayı sağ tıklayıp "Gönder*Notepad" yapabilirsiniz. Crimson Editor ise güzel bir not defteri alternatifi. Kod renklendirme, bol geri alma, büyük dosyaları kolay açma gibi özellikleri bağımlılık yapabilir.

Örnek kodlar ne ile çalışacak?

- Assembler (*.asm) : Tasm.exe, TLink.exe ve bunlar ait dpmiload.exe ve dpmimem.dll. Borland firmasına ait olan Turbo Pascal veya Turbo C'niz var ise bu dosyaları "\\Bin" dizininde bulabilirsiniz. CD'mizde de "Ornekler\Asm" dizininde bunlar var. Bu arada "Windows\Command" dizinindeki debug programı da *.com dosya oluşturmamıza yardımcı oluyor. Editör olarak not defteri kullanacağız. {DOS programcılığı}
- Basic (*.bas) : Gwbasic.exe sadece satır numaralı olan *.bas dosyaları okuyabiliyor. Genellikle QuickBasic, VB dilinin *.bas dosyaları hariç tüm *.bas dosyaları işler. {DOS programcılığı}
- Visual Basic (*.mak veya *.vbp - proje dosyası, *.frm - kodlar, *.bas - modül bilgileri) : VB4 ve üzeri yeterli. {Windows programcılığı}
- C (*.c, *.cpp) : Turbo C veya Borland C gerekli. {DOS programcılığı}
- Pascal (*.pas) : Turbo Pascal veya Borland Pascal gerekli. {DOS programcılığı}
- Delphi (*.dpr - proje dosyası, *.pas - unitler bu dosyalarda, *.dfm - form bilgileri) : Delphi 3 veya üzeri gerekiyor. {Windows programcılığı}
- Excel (*.xls) : Tabii ki Office 9x ve üstü ile makroları işleyebilirsiniz. {Windows programcılığı}
- Flash (*.fla, *.swf) : Flash ile tasarım yapılabilir, Internet Explorer ile ön izleme yapabilirsiniz, isterseniz exe olarak da sonuç elde edebilirsiniz. {Windows programcılığı}
- VB Script (*.vbs) : WSH (Windows Scripting Host - Windows Komut Dosyası Sistemi) kurulu ise soyadı *.vbs (Visual Basic Script) veya *.js (J-Script) olan dosyalar çalıştırılabilir haldedir. "Windows\Samples\WSH" dizinindeki örneklerle bir göz atın. Not defteri ile düzenlenebiliyorlar. {Windows programcılığı}
- Web sayfası (*.htm, *.html, *.asp, *.php) : Not defteri veya FrontPage, Dreamweaver gibi programlar ile tasarlanıp, Internet Explorer, Mozilla veya Netscape ile izlenebilir. *.asp için ayrıca IIS kurmak gerekiyor. *.php için de Xampp tavsiyem. {Internet programcılığı}
- VRML (*.wrl) : Virtual Reality Markup Language. Görüntülemek için cortvrml.exe gerekiyor. Internet Explorer'da görebilirsiniz sonuçları. "cd2utility\VRML" dizininde editör örnekleri bulabilirsiniz. isa10.zip'i tavsiye ederim. {Internet programcılığı}

Eğer hala değişik uzantılı dosyalar görürseniz, şimdilik onlar bizim işimize yaramıyor. Mesela *.bak, *.~dp gibi dosyalar programlama dilinin kendi yedek ve yardımcı dosyaları.

Siz de sonuçta **sistem yazılımı** - system software (bilgisayarın çalışması ile ilgili işletim sistemi gibi), **mesleki ve ticari yazılımı** - application software (muhasabe, kelime işlemci gibi kullanıcılara yardımcı), **yardımcı programlar** - utility (CD Player, ince ayar programı, mp3 çalıcı gibi işlemler yapan), **görsel yazılım** - animation (oyun programlama, eğlence), **yapay zeka** - artificial intelligence (düşünen, akıllı robotlar) üzerine bir program yazmanız gerekebilir. *Herhangi* bir dilde bu isteğinizi başarabilirsiniz. Unutmayın ki kullandığımız tüm programlar ve işletim sistemleri bir programlama dili ile programcılarının elinden geçmiş yazılımlardır. O kadar, uzaylıların bu işlerle ilgileri yok... Takım çalışmasını da unutmamak gerekli. Örneğin tek başınıza bir oyun programının müziğini, seslendirmesini, hikaye akışını, harita, nesne tasarımı, yardım ve kurulumunu yapmanız zor değil mi? Elinizden geldiğince proje arkadaşları edinip kendinizi ve takımınızı geliştirin. Her ne kadar bencillikte bizi kimse geçemese de, takım halinde olmanın verdiği zevk tek başına çalışmaktan çok çok daha fazla.

Şu adımlar da program geliştirme sırasında yaşadığımız aşamalardır:

1. Daha iyi tasarım ve geliştirme için programa başlamadan önce mevcut sistemi analiz ederek, altyapı hazırlanır.
2. Problemin tanımı yapılır, sistem elemanları ile bu elemanlar arasındaki ilişki akış diyagramları ile belirtilir.
3. Sistemdeki hazır bilgiler, kullanıcılar için gerekli çıktılar, teknik, ekonomik araştırma bilgileri değerlendirilerek programa başlanır.
4. Girdi ve çıktılar etkin kullanım için tasarlanır, kullanıcı arabirimi iyileştirilir.
5. Yapılan program test aşamasında ihtiyacı görüp görmediği belirlenir.

6

6. Yeni isteklere ve eski sistemlere olan uyumluluk kontrol edilir.
7. Program paketlenir ve yardım kısmı, dökümantasyon işlemleri yapılır.

Her an aklımızda olması gereken kullanıcının programınız karşısında verebileceği tepkilerdir. Eğer kullanıcı dostu olmak yerine kendi yapınızda ısrar ederseniz sonunuz iyi olmayabilir. Genel standartlara ve prensiplere uymaya çalışmak en iyisidir.

Yazar Hakkında

1975 Kırşehir Kaman doğumluyum. Ne Amiga ile ne de Commodore ile uğraştım. Windows 3.1 bile doğru düzgün kullanmadım, Windows genel platformum. Linux (*linuks, linaks, laynaks diye okunur*) ve BeOS (*bi o es diye okunur*) kurma denemelerim oldu ama en iyisi şimdilik Windows XP işletim sistemi. Kurulumu kolay, oyun (DirectX) ve görsel programlama için uygun. Aslında bu sözleri biraz içim burkularak söylüyorum, çünkü keşke Linux veya Unix için bunları söyleseydim.

Bir ara elime bir el bilgisayarı olan Casio FX-880P geçmişti. İçindeki 64KB belleğine fazla uzun bir şey yazılamıyordu, ama yine de Basic benzeri dili ile Shell adında Türkçe programlama dili, Çarkıfelek, Sayı bulma, Batak kağıt gibi oyunlar yapmıştım. Kitabının dökümantasyonu çok hoşuma gitmişti. Sade ve tüm komutları örnekle anlatma.

1991'den bu yana, liseden itibaren programlama ile ilgileniyorum. Windows programcılığı ile 1997'de Visual Basic ile başladım. Asıl ilgi alanım Delphi olduğu için daha çok bu dilde programlarımı size tanıtacağım. İlk öğrendiğim programlama dili lisede (Balıkesir Anadolu Ticaret Meslek Lisesi) 1991'de GWBasic. Yani altyapı olarak Visual Basic'ci olmam gerekirken Delphi beni mıknaş gibi çekti.

1993'de Marmara Teknik Eğitim Fakültesi İngilizce Bilgisayar Öğretmenliğini kazanarak 5 yıllık eğitimde sırasıyla QuickBasic, Pascal, C, Assembler (8088), PLC, dbase, microcontroller (8051), ve VB3.0 öğrendik. Son yılda 1998'de yani tez ortağım Batur Baykan ile Delphi 3.0'da tez hazırlamaya karar verdik. Birkaç ay içinde Okul Ders Programı adlı program ile cebelleşmeye başlamıştık ve bir yıl boyunca (yaklaşık 750 saat) birçok şeyi zevkle çözdük. Burada "team work" yani takım çalışması ile ilgili olarak size diyebileceğim, ortak ile çalışmanın tek başına iş yapmaktan daha zevkli ve faydalı olduğudur.

Okuldan sonra da öğretmen olunca öğrendiğim dillere pek yenisini eklemedim. C++, Java, veya C# öğrenmek şimdilik benim için çok ağır. Delphi ile Visual C'yi karşılaştırmayı pek gerek yok, ama Delphi de VC kadar zengin kütüphanelidir. PIC üzerine uğraşlarım devam ediyor.

Seviye olarak okuyucuların orta seviyede olması gerekiyor, birkaç dilde bir şeyler yazmış olmanız gerek. Yani size sıfırdan bütün ayrıntıları ile bir dilde bir program nasıl yazılır anlatmayacağım. Bunun için **sevdiğiniz dili** belirleyip ona özel kitabı almanız veya kurs / okuldan bunu edinmelisiniz. Aslında "Hangi dili öğrenmeliyim?" sorusuna cevap vermeye çalışacağız. C ve benzeri, sadece programcılığa giriş dili olmalı bence. Daha sonra da VB, Delphi veya VC'ye geçilir. Aslında Basic'den uzak durulması da tavsiye ediliyor. Çünkü insanlara kötü alışkanlıklar kazandırıyor.

Görsel programlama ile kod yazmaktan kurtulunuyormuş gibi zannedilse de iyi bir program yazdığınızda belki programınız yüzlerce binlerce satırdan oluşacak. Daha ilerisi de VB.NET gibi internet tabanlı Windows programlarına benzeyen ve veritabanı kullanan programlar olacaktır.

Site : www.yunus.projesi.com
Mail : tbagriyanik@mynet.com
Şehir : İstanbul / Türkiye

Başlangıç Tarihi	: 21 Haziran 2001 Perşembe
Son güncelleme	: 2 Haziran 2009 Salı
Toplam harf sayısı	: 639889 adet
Toplam kod satırı	: 12000 üzeri
Toplam kaydetme	: 275 defa
Toplam süre	: 9,38 saat
Resim sayısı	: 228 adet (7,2 MB – resimler "dosyaya bağlı" şekilde eklendi)
Tasarım araçları	: Microsoft Windows XP Türkçe SP2, Delphi 7, Delphi2005, Visual Basic 6, VB.NET2003, Borland Pascal 7.0, Turbo C++ 3.0, Qbasic 4.5, Microsoft Office 2003 Türkçe, Adobe Photoshop CE, Macromedia Dreamweaver 8 ve Flash 8
Donanım	: P4 2400, 512 DDRAM, 160 GB, 128 MB GeForceFx5200, 17" LG 775FT, USR ADSL, Dell Inspiron 6000 notebook

Yazılım Geliştirme ve Böcek Ayıklama



İyi bir programın temel özellikleri; bunların sırası sizin stilinizi belirler:

- 🖥️ **Doğruluk** : Verilen görevlerin tam olarak yerine getirilmesi.
- 🖥️ **Dayanıklılık** : Beklenmedik hatalardan dolayı programın çalışması kesilmemeli.
- 🖥️ **Genişleyebilirlik** : İleri aşamalarda görevlerin değişikliği veya yenilerinin eklenmesi kolay olmalıdır.
- 🖥️ **Basitlik** : Karmaşık tasarımlardan kaçınmak gerekir.
- 🖥️ **Modülerlik** : Program kodları başka programlarda da kullanılabilir.
- 🖥️ **Uyumluluk** : Başka bilgisayar ve sistemlerde çalışabilir.
- 🖥️ **Kontrol edilebilirlik** : Hata olabilecek yerlere açıklayıcı hata mesajları konulmalıdır.
- 🖥️ **Kolay kullanım** : Kullanıcı arabirimi kolay olmalı ve rahat öğrenilebilir.
- 🖥️ **Parçalanabilirlik** : Problemi küçük parçalara ayrılarak yazmak.
- 🖥️ **Anlaşılabilirlik** : Başkasının yazdığı program elden geçirilirse rahatça okunabilir.
- 🖥️ **Koruma** : Modüller birbirlerine müdahale etmemelidir.



Bir program yazarken aşağıdaki öneriler işinize yarayabilir:

- 🖥️ Yazılımınız ne kadar büyürse o kadar içinde bug – **böcek** bulunma ihtimali artar.
- 🖥️ Böcek oluşmasını nasıl önleyebileceğinizi, otomatik olarak nasıl yakalayabileceğinizi düşünmelisiniz.
- 🖥️ Algoritmanızı oluştururken, kullanacağınız dili tam olarak öğrenerek koşullarınızı tamamlayınız.
- 🖥️ Birden fazla yol göz önüne alarak alternatifler ekleyin.
- 🖥️ Çalışma anında beliren böcek sinyallerini göz ardı etmeyin. Kendi kendilerine kaybolmazlar. Bunu beklerseniz onlar daha da büyürler.
- 🖥️ Değişkenlerinizin değerlerini takip edin ve aykırı değerler almaları ne zaman ve nerede oluşuyor debug – böcek ayıklama ile görün.
- 🖥️ Eğer programınızı kontrol amacı ile adım adım işlemekten çekinirseniz, tahmin etmediğiniz hatalarla kullanıcı muhatap olmak zorunda kalır.
- 🖥️ Denetlenmesi kolay adlandırma ve algoritmalar oluşturun.
- 🖥️ Hızlı çalışan program yapmak için dayanıksız kodlar oluşturmayın.
- 🖥️ Belki tasarımınızı baştan tekrar oluşturmanız gerekebilir ama yazılımınızı her zaman test edin.
- 🖥️ Kestirme ve pratik programlama risklidir. Kendi ihtiyacınıza göre çözüm üretin.
- 🖥️ Ara sıra temizlik yaparak, artık yani artan kodlarınızın böceğe sebep olmamalarını sağlayın.
- 🖥️ Goto deyiminden kaçınmak, programın karmaşık dallanmalarını önler ve derleyicinin optimizasyonunu artırır, fakat yine de **yerinde** kullanılan goto ile kod daha anlaşılır ve verimli olabilir.
- 🖥️ Yaptığınız değişiklik çok küçük olabilir, ama hata oluşturabileceğini de unutmayınız.
- 🖥️ Sürümler oluşturarak ilerleyin. Son ticari halinden hata ayıklama kodlarınızı çıkararak küçülterek ilgililere teslim edin.
- 🖥️ Program içine yerleştirdiğiniz açıklama satırları az ve öz olmalı, ya da hiç yazmayın.
- 🖥️ İmkansız olan bir girdi olabilir mi? Kullanıcı her zaman sizin gibi düşünmez. Bu ihtimallere de önlem almalısınız. Sorunları sessizce çözen kodlar yazabilirsiniz.
- 🖥️ Böceği gizlemekten çok onu yok etmeye çalışın.
- 🖥️ Farklı algoritma kullanarak böceklerin sebebini çözebilirsiniz.
- 🖥️ Devamlı hata mesajları ile kullanıcıyı telaşlandırmayınız.
- 🖥️ Böceği açığa çıkarmanın yöntemi, onu bulduğunuz anda ortadan kaldırmaktır. Böcekleri tekrarlamaya zorlayın.
- 🖥️ Yolları kesmekle şehirdeki eşkıyaları yok edemezsiniz, sessizce bekliyorlardır. Evleri tek tek aramak eşdeğeri olan, programınızda böcekleri aktif olarak arayan kodlar ekleyebilirsiniz.
- 🖥️ Çok az sayıda programcı kodlara kesme noktaları ekleyerek, kod içinde adım adım ilerleme ve girdilerin çıktıyla dönüştüğü yerlere bakma alışkanlığına sahip. Bu işlem angarya değildir.
- 🖥️ Kodunuzda adım adım ilerlerken veri akışına odaklanın.
- 🖥️ Arabiriminizdeki aksaklıkları bulup yok edin.
- 🖥️ Kaybolan bir böceği göz ardı etmeyin, gizleniyor olabilir.
- 🖥️ Zamanında bulunan böcek daha fazla böcek oluşmasına engel olur.
- 🖥️ Belirtiyi değil, oluşum nedenini düzeltin.
- 🖥️ Aptal böcek yoktur. Asıl sorun programcının onu bulamamış olmasıdır.
- 🖥️ Aynı böceğin sizi iki kere sokmasına izin vermeyin!



Bilgisayar kullanımı ile ilgili bir kaç ipucu:

- 🖥️ Windows'unuz devamlı göçüyor, mavi ekranla karşılaşıyorsanız, bunun sebepleri ya o sırada çalışan hatalı bir program yada ucuz donanım parçalarıdır. Markalı RAM ve iyi bir anakart alın.
- 🖥️ Antivirüs programınızı devamlı çalışması gerekli. Klasöre ve dosyaya sağ tıklayıp da "Virüs kontrolü yap" diyebilen antivirüs yazılımı kullanabilirsiniz.

- Masaüstünde resim kullanmak yerine teal – yeşil, siyah veya koyu mavi gibi tek renk kullanmanız genel performansı olumlu etkiler. Yazıların okunaklı renklerde olması gözü en az yoran faktördür. (sarı üzerine siyah gibi...)
- Masaüstündeki gereksiz kısayolların hepsini silin. Böylece Windows bir de onları güncellemekle uğraşmaz. Zaten Windows Gezgin (Win+E) yeterince kullanışlı bir programdır.
- Dosya uzantılarının "Klasör Seçenekleri*Görünüm*Bilinen dosya türlerinin dosya uzantılarını gizle" seçeneğini iptal ederek gözükmelerini sağlayın.
- Klavye kısayolları iyi bilgisayar kullananların en büyük yardımcısıdır. Fare yerine Alt+Tab, Win+D, gibi kısayolları kullanmak daha hızlıdır. Kullandığınız paket programların da kısayollarını en kısa(!) zamanda öğrenin.
- Bir programı kullanmayı seviyor olabilirsiniz, ama alternatif programları da muhakkak deneyin. Yine geri döneceksiniz, fakat bazen işe yarıyor.
- Her şeyin son versiyonunu kullanmak zorunda değilsiniz. Elinizdekileri en iyi şekilde kullanma çalışın.
- Muhakkak elinizin altında İngilizce sözlük bulunsun. WordWeb32, Moonstar programı çok kullanışlı.
- Devamlı hafızada yer işgal eden gereksiz bütün programları açılıştan – "Başlat*Çalıştır... *msconfig" ile iptal edin.
- Eğer güzelce çalışan bir donanımınız var, ve siz de "Dur şunu daha hızlandırayım!" diye yola çıkmışsanız, büyük bir ihtimalle elinize mahvolmuş bir Windows veya bilgisayar geçer. Özellikle BIOS ve registry – Windows Kayıt Veritabanı ayarları çok tehlikelidir.
- Donanım upgrade – güncelleme işlemi ancak zamanında yapılırsa güncellemedir, yoksa para kaybıdır.
- Bilgisayarda radyo dinlemek, televizyon seyretmek güzel şeyler, fakat en iyisi normal radyo teyp ve televizyondan yararlanmak. MP3 dinlemek ve Winamp Shoutcast serbest neyse ki...
- İnternet'e firewall – ateş duvarı programınızı çalıştırmadan çıkmayın. Norton Internet Security, ZoneAlarm ve Tiny Personal Firewall tavsiyem.
- İnternet'den sadece chat ve gereksiz siteleri dolaşmakla faydalanmayın.
- Vee oyunlar. Muhakkak oyun ile stres atıyorsunuzdur. Ama işyeri ile evi birbirine karıştırmayın.
- Bilgisayar kullanırken boyun, el, bilek, sırt ağrıları çekiyorsanız, bunların kronikleşmesine izin vermeden sebeplerini giderin. Düzgün masa, sandalye, ergonomik klavye ve fare, rahat görebileceğiniz monitör ve ekran çözünürlüğü gibi...
- Bilgisayarınıza ne kadar özen gösteriyorsanız aynı şekilde çevrenizin de temiz ve düzenli olmasını sağlayın. Kitaplarınızı, CD/DVD 'lerinizi yıpranmaktan koruyun. Park ve bahçeleri de...



Murphy'nin Yasaları:

- Ters gidebilecek bir şey mutlaka ters gidecektir. Hatta tersliğin en az mümkün görüldüğü anda bile.
- Siz süpermen değilsiniz.
- Aptalca görünen bir şey işe yarıyorsa, o şey aptalca değildir.
- Emin değilsen silahını dolu kabul et.
- Eğer saldırı iyi gidiyorsa, pusuya düşmüşsündür.
- Tüm beş saniyelik el bombaları üç saniyeliktir.
- Kolay yol her zaman mayınlanmıştır.
- Önemli olan şeyler daima basittir.
- Basit olan şeyler daima zordur.
- Eğer düşman menzil içinde ise sen de öylesindir.
- Dost ateşi dostun ateşi değildir.
- Yapabildiğin her şey vurulmana yol açabilir, hiç bir şey yapmaman da dahil.
- Üstüne gelen düşman ateşinden daha isabetli olan tek şey, üstüne doğru gelen dost ateşidir.
- İki taraf da kaybetmeye başladığına inandığı anda, ikisi de haklıdır.
- Profesyonellerin ne yapacağını kestirebilirsiniz, ama dünya amatörlerle doludur.
- Süngü kanunu der ki, mermisi olan kazanır.
- Kusursuz planlar, kusursuz değildir.
- Savaşta üniforması daha gösterişli olan taraf kaybeder.
- Eğer düşmanı görebiliyorsan o da seni görebilir.

Bölüm 1 – Assembler

Tüm dillerle doğal olarak uyumlu olan mikroişlemci programcılığı bazıları için çok ilgi çekici olabilir. Hata mesajlarını dikkatli okumanız gerekiyor. MS-DOS da hiç hata mesajı görmeden makineniz kilitlenebilir. Genellikle eliniz kolunuz bağlı gibi hissedebilirsiniz. Yani mikroişlemci programcılığı bir fili emekleyerek veya sürünerek avlamaya benzer. Yine de avantajları da var. İleri bölümlerde Delphi ve diğer diller içinde bile hala Assembler kodları kullanıldığını görebilirsiniz. Çünkü çok az kod yazarak donanım (grafik ekrana geçmek, paralel bağlantı noktasına sinyal yollamak, ...), işlemciye erişebilir ve hızlı (sürünerek de olsa!?) çalışan program yazabilirsiniz.

Alt katmanında ikilik – binary (*baynırı diye okunur*) ve onaltılık – hexadecimal (*heksadesimil diye okunur*) sayı sistemi vardır. Yani sırf 1 ve 0'larla da yani bitlerle program yazılabilir. Zaten bu programlama diline makine dili programlama da deniliyor. Bu konuya yakın olarak biraz elektronikten bahsetmek gerekirse, eğer bir buzdolabı gibi yıllarca hep aynı işlemleri yapacak bir devre tasarlama ihtiyacı duyarsak, hedeflediğimiz giriş ve çıkış değerlerini bir ROM – Read Only Memory salt okunur belleğe yükleyerek işimizi görürüz. Tabii hepimizin evinde böyle bir EPROM – erasable programmable read only memory silici ve yazıcı olmadığına göre, bu tip programlama sadece sanayi veya fabrika ortamında yapılabilir. Novell Netware kullananlar bilirler, ağ kartlarına boot – açılış EPROM takılarak terminalerin açılmalarını sağlanabiliyordu. Ayrıca hala bilgisayarlarımızın ilk donanım bilgileri BIOS çipinde saklanıyor. BIOS da bir EPROM'dur. Açılışta BIOS bilgileri daha hızlı olan RAM'in ilk megabyte'ının üst kısmına yerleşir. Ayrıca ilk megabyte görüntü belleği, Expanded – genişletilmiş; kalan tüm RAM ise Extended – uzatılmış bellek olarak ayrılır.

PLA – Programmable Logic Array ile sadece and, or gibi devre elemanları ile istenen giriş değerleri ile istenen çıkışlar elde edilir. Logic gates – mantık kapıları kullanarak Flip – Flop denilen en küçük, yani bir bit saklayabilen hafıza devresi yapılır. Toplayıcı, çıkarıcı, karşılaştırıcı, encoder, decoder, multiplexor (çoğullayıcı – mux), demultiplexor, sayaç da mantık kapıları kullanılarak yapılabilir.

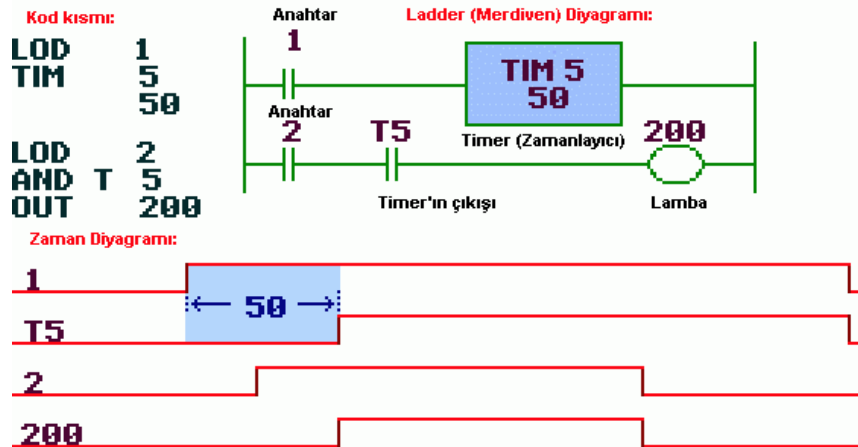
And (Çarpma)		
A	B	Sonuc=A.B
0	0	0
0	1	0
1	0	0
1	1	1

Or (Toplama)		
A	B	Sonuc=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Not (Tersleme)	
A	Sonuc
0	1
1	0

XOr (Özel OR)		
A	B	Sonuc=A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

"CD2\programlama\pic\PLC" dizininde Programmable Logic Controller – programlanabilir mantık denetleyicisi (microcontroller) ile ilgili dosyalar bulabilirsiniz. PLC ise PLA'nın gelişmiş halidir. Mikroişlemci kodları gibi satır satır programlanabilen, bilgisayarın çok küçültülmüş halidir. Ama mantık olarak bilgisayar aynı anda tek bir işlem yürütebilirken, PLC giriş bilgilerini *parallel* olarak işleyip çıkış darbeleri üretiyor. DC – direct current girişleri (24 Volt) vardır ve bu bilgileri programcı ladder diagram – merdiven diyagramı ile tasarlayıp istenen çıkış darbelerini çıkış rölelerine (220 Volt) aktarır. Eğer biraz elektrik bilginiz varsa resimdeki giriş çıkışları rahatlıkla anlayabilirsiniz. Eğer bir nolu anahtara basılırsa zamanlayıcı aktif hale gelip, 50 saniye sonra çıkış veriyor ve eğer iki numaralı anahtara da basılırsa lambamız yanıyor. Zaman diyagramında 200 çıkışının sadece 2 ve T5 yüksek değerde iken çalıştığını görebilirsiniz.



Resim 1 – idec FA-1j uygulaması

Bilgisayarımız transistör, direnç, kondansatör gibi binlerce elektrik ve elektronik parçalardan oluşuyor. En temel değer elektrik var "1", elektrik yok "0" 'dır. Yani oyunlar, müzik parçaları, programlar tamamen binary sistem üzerine kuruludur, diğer deyimle dijitaldirler.

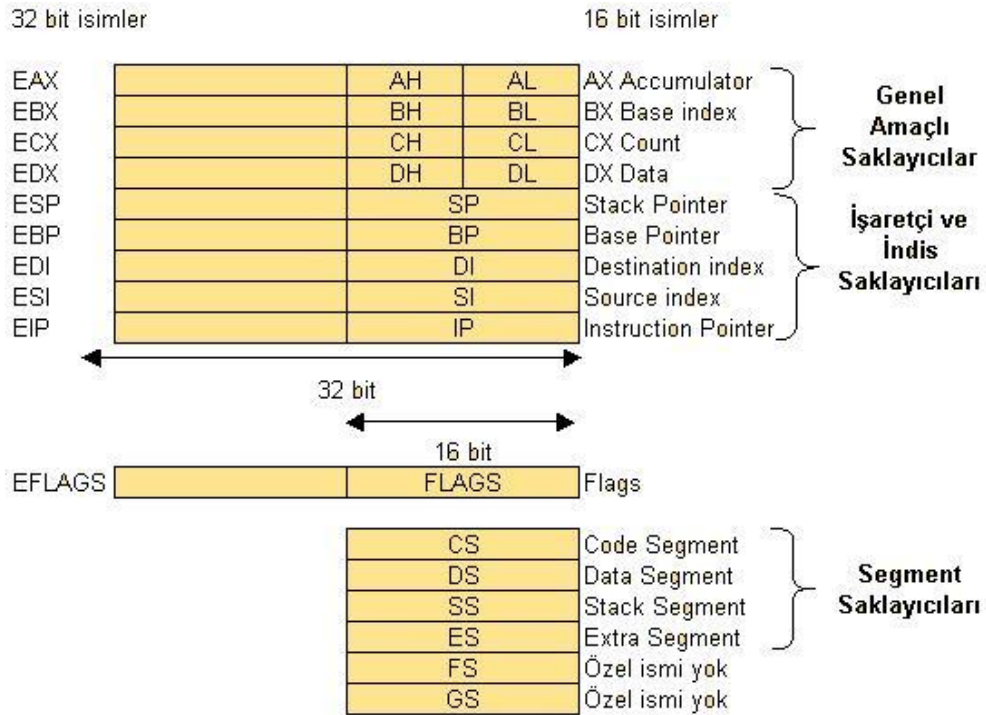
Bit kavramını biraz inceledikten sonra şimdi de biraz işlemcinin iç yapısında bulunan register – saklayıcıların yapısına bakalım. Resimde 32 ve 16 bit deyimleri ile bilgisayarımızın eğer Pentium ise 32 bit, intel286 ise 16 bit işlemci olduğunu anlıyoruz. Yakın gelecekte (2006) 64 bit işlemcilerimiz ve işletim sistemlerimiz olacak. Bu hız ve performans ile ilgili. Genellikle 32 bit yazılım 16 bit yazılımdan daha hızlı çalışır.

En basit saklayıcılar 8 bit olan AL, AH, BL, BH, CL, CH, DL ve DH'dir. Tüm matematiksel ve diğer işlemler AX, BX, CX ve DX üzerinde yapılır. SS:SP, CS:IP, DS:SI ve ES:DI birlikte kullanılarak adres belirtirler. Segmentler 64KB yer kaplayan özel alanlardır. Örneğin CS 1A00H değerini, IP de 0200H değerini içeriyorsa iki yazmacı birleştirerek yani normalleştirerek 1A200H adresini elde ederiz. CS ve IP 16 bit iken, sonuç 20 bit olması için CS sonuna 0 eklenip IP ile toplanır. 8086 işlemcisinin 20 bit address bus – adres hattı kapasitesi ile 2^{20} 1MB hafızayı adresleme imkanı vardır. AX yazmacının 80386 ve üzeri için gelişmiş 32 bit hali EAX'dir. Buradaki E – extended uzatılmış anlamındadır.

1970 yılında çıkan 4040 ve 8008 intel çipleri sadece hesap makinelerinde kullanıldı. Bunlar sadece 64 KB adresleyebilen sekiz bit işlemcilerdi. Şu anda kullandığımız intel işlemcilerimiz intel286 hatta 1978'de piyasaya sürülen 16 bitlik 4.77 MHz 8086 ile geriye uyumlu çalışmak zorunda olduğu için yazmaç yapıları hala 16 ve 8 bit olarak da çalışıyor. Yeni bir programın eski bir bilgisayarda çalışabilmesine hayret etmemek gerek. Fakat Pentium işlemcilerdeki MMX, SSE, SSE2 ve SSE3(AMD'de 3DNow!) komut kümeleri geriye doğru uyumlu değildir.

CISC – Complex Instruction Set Computer (Karmaşık komut kümeli bilgisayar) yöntemi ile programlandığı için intel, geniş komut kümesi ile esnek fakat hızı düşük işlemcidir. PowerPC gibi RISC – Reduced Instruction Set Computer (azaltılmış komut kümeli bilgisayar) olan sistemler ise belli işlemleri çok hızlı bir şekilde yapabilen işlemcilerdir, genellikle üzerlerinde özel işletim sistemleri çalışır. Ama artık Mac işletim sistemi intel uyumlu.

Resim 2'deki gibi yazmaçlarımız bu kadar. Ama işlemci içinde sadece bunlar yok. Cache – ön bellek, FPU – floating point unit, giriş / çıkış yol arabirim mantığı, code cache – kod ön hafıza, tamsayı yürütme birimleri, komut kod çözücü, ve buna benzer ünitelerden oluşur. İşlemcinin anakart bağlantıları da içindekine benzer. Dış cache, RAM – Random Access Memory, sistem veri yolları, giriş / çıkış yolları, disk kontrolcüsü gibi bölümlere bağlanır ve onları yönetir. Donanım ile ilgili bazı örneklerimizi ilerleyen bölümlerde bulabilirsiniz.



Resim 2 – intel x86 ailesinin işlemci yazmaçları

Örnek mikroişlemci komutları:

mov ah, bh	8 bit (1 Byte) bh içeriği ah'ye kopyalanır
mov [bx], ax	16 bit ax yazmaç değeri bx adresine kopyalanır
jmp ax	ax adresine dallan
push ax	ax'i yığında sakla
pop bx	yığına son atılan değeri bx'e aktar
add al, cl	al=al+cl
inc bp	bp=bp+1
dec bp	bp=bp-1
cmp al, cl	sadece bayrakları etkileyen al-cl çıkarma işlemi
mul dl	ax=ax*dl
div cl	al=ax/cl , kalan da ah yazmacında
and bl, 7	0111 ile bl maskeleniyor (çarpma), 4. bit sıfırlanıyor
or bl, 2	0010 ile bl'nin 2. biti 1 haline getiriliyor
xor al, al	al içi sıfırlanıyor, temizleniyor
not cl	cl içi tersleniyor
shl cx	cx'in içi sola 1 bit kaydırılır (yani 2 katı alınır)
shr cx	cx'in içi sağa 1 bit kaydırılır (yani 2'ye bölünür)
rol di	di'nin soldaki ilk biti sona, kalan bitler sola kayar
loop ETIKET	ETIKET adresine cx 0 oluncaya dek döner (for gibi)
lahf	ah'ye flag – bayrak bilgilerini saklar
sahf	ah'den flag – bayrak yazmacına geri yükler
pushf	bayraklar stack segment'e itilir
popf	bayraklar stack segment'ten geri alınır

mov ax, 2000	ax = 2000
mov bx, 1000	bx = 1000
cmp ax, bx	ax ile bx'i karşılaştır
jb kucuk	eğer ax küçükse kucuk etiketine git
buyuk: mov dx, ax	dx = ax
sub dx, bx	dx = dx - bx
jmp tamam	koşulsuz tamam etiketine git
kucuk: mov dx, bx	dx = bx
sub dx, ax	dx = dx - ax
tamam: nop	işlem yok

Yukarıdaki çıkarma işlemi yapan örneği "Başlat*Çalıştır...*debug" programı ile şöyle yazabilirsiniz: (Ornekler\Asm\asm.zip*cikarma.txt). Delphi dilinde ise adres belirtmeye gerek kalmadan @label şeklinde etiketler ile loop, call ve jmp komutlarını kullanabiliriz. (Ornekler\Asm\delphi_asm.zip*Project2.dpr) Diğer örneklerde debug'ın komutlarını bulabilirsiniz.

```
-a
18A2:0100 B80020      MOV    AX,2000          ; 18A2 adresi sizde farklı olabilir
18A2:0103 BB0010      MOV    BX,1000
18A2:0106 39D8        CMP    AX,BX
18A2:0108 7206        JB     0110             ; küçükse (jump if below)
18A2:010A 89C2        MOV    DX,AX
18A2:010C 29DA        SUB    DX,BX
18A2:010E EB04        JMP    0114             ; son
18A2:0110 89DA        MOV    DX,BX
18A2:0112 29C2        SUB    DX,AX
18A2:0114 90          NOP
-t 6                  ; 6 satır trace – izle yani çalıştır
```

```
AX=2000 BX=1000 CX=0000 DX=F500 SP=FFEE BP=0000 SI=0000 DI=0000
DS=18A2 ES=18A2 SS=18A2 CS=18A2 IP=0103 NV UP EI NG NZ NA PE CY
18A2:0103 BB0010      MOV    BX,1000
```

```
AX=2000 BX=1000 CX=0000 DX=F500 SP=FFEE BP=0000 SI=0000 DI=0000
DS=18A2 ES=18A2 SS=18A2 CS=18A2 IP=0106 NV UP EI NG NZ NA PE CY
18A2:0106 39D8        CMP    AX,BX
```

```
AX=2000 BX=1000 CX=0000 DX=F500 SP=FFEE BP=0000 SI=0000 DI=0000
DS=18A2 ES=18A2 SS=18A2 CS=18A2 IP=0108 NV UP EI PL NZ NA PE NC
```

18A2:0108 7206 JB 0110

AX=2000 BX=1000 CX=0000 DX=F500 SP=FFEE BP=0000 SI=0000 DI=0000
DS=18A2 ES=18A2 SS=18A2 CS=18A2 IP=010A NV UP EI PL NZ NA PE NC
18A2:010A 89C2 MOV DX,AX

AX=2000 BX=1000 CX=0000 DX=2000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=18A2 ES=18A2 SS=18A2 CS=18A2 IP=010C NV UP EI PL NZ NA PE NC
18A2:010C 29DA SUB DX,BX

AX=2000 BX=1000 CX=0000 DX=1000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=18A2 ES=18A2 SS=18A2 CS=18A2 IP=010E NV UP EI PL NZ NA PE NC
18A2:010E EB04 JMP 0114
-t

AX=2000 BX=1000 CX=0000 DX=1000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=18A2 ES=18A2 SS=18A2 CS=18A2 IP=0114 NV UP EI PL NZ NA PE NC
18A2:0114 90 NOP

AX BX'den büyük olduğu için 2000-1000 sonucu olan 1000 değeri DX yazmacında saklanıyor. İsterseniz AX'e 2001, BX'e 1975 değerini vererek de deneyebilirsiniz. Sonuç neden 068C oldu da 26 olmadı acaba? Onaltılık sayı sisteminde olduğumuzu unutmayalım. "Ornekler\Asm\delphi_asm.zip*Project2.dpr" 'de ise sonuç onluk sisteme çevrildiğinden dolayı 26 elde edebilirsiniz.

-r ip çalışacak programın başlangıç adresi 100 olmalı
IP 0114
:100
-a 100 sadece iki satır değişecek
18A2:0100 mov ax,2001
18A2:0103 mov bx,1975
18A2:0106
-t 6
...

Aşağıda bazı örnekler ise *.asm uzantılıdır. Bunları derlemek için CD'deki "Ornekler\Asm\tasm.zip" dizindeki dosyaları (Tasm.exe, Tlink.exe, dpmload.exe, dpmimem.dll) sabit diskinize kopyalayın. Not defterinde örnek kodları yazıp kendi diskinizdeki bu dizine kaydedebilirsiniz. Şimdi yapacağınız "exeyap.bat" a kısayol oluşturup, benim yaptığım kısayoldaki gibi, kısayola sağ tıklayıp "Özellikler*Program*Komut Satırı:exeyap.bat ?" haline getirin. Bu "?", exeyap.bat dosyasının çalışmadan önce bize parametre ile dosya adını sormasını sağlayacak. Kısayolu çalıştırın ve "dosyaadi" yazıp Enter'a basın. İşte bitti, hemen hemen hiç MS-DOS komut isteminde bir işimiz olmadı. exeyap.bat ve comyap.bat dosyalarının içeriği şöyle:

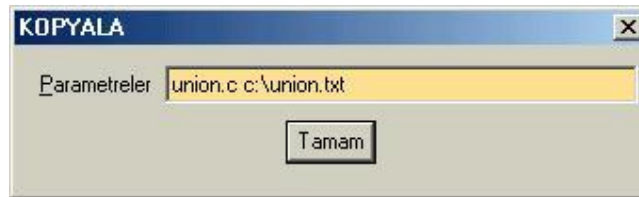
```
@echo off
tasm %1
tlink %1
erase *.map
erase *.obj
%1.exe
```

exeyap.bat olarak kaydedin

```
@echo off
tasm %1 /m
tlink /t %1
erase *.map
erase *.obj
%1.com
```

comyap.bat olarak kaydedin

%1 deyimlerinin anlamı *.bat dosyaya parametre eklenebilmesi içindir. Bu sayede "c:\>exeyap.bat deneme.asm ↵" komutu ile bir anda programımız eğer hata yoksa derleniyor. Windows'ta da uygulamaların kısayollarına ? ekleyerek açılmalarından önce bize küçük bir pencerede parametre girişi imkanı verebiliyoruz:



1. Mavi.txt (mavi renkli ekran)

(Ornekler\Asm\Diger.zip*Mavi.TXT)

İlk örneğimiz “Mavi.txt” dosyasındaki örnek. Aşağıdaki kodları not defteri ile yazabilirsiniz.



Resim 3 – Mavi.com

Doğrudan CD'mizden denemek için, bu örneği not defteri ile açıp hepsini seçin ve kopyalayın. "Başlat*Çalıştır..." 'dan "debug" (*dibag diye okunur*) komutu verin. Çıkan siyah komut isteminin pencere halinde olması için Alt+Enter yapıp, araç çubuğundaki "Yapıştır" ile hafızadaki bilgilerin yapışmasını seyredin. İş bitmiştir, ilk programınızı derlediniz. Masa üstündeki mavi.com'un tek yaptığı şey mavi bir metin ekran oluşturmak.

? işareti ile de isterseniz debug'ın yardımına bakabilirsiniz.

Kod yazma kısmı:	Açıklama kısmı:
-a 100	100'ncü adresten itibaren kod yazıyoruz (a - assemble)
mov ah,12	
mov bl,10	
int 10	
mov ax,40	
mov ds,ax	
mov ax,600	
mov bh,1f	
SUB CX,CX	
mov dh,18	
cmp bl,10	
jz 11d	
mov dh,[84]	
mov dl,[4a]	
dec dl	
int 10	
mov ah,2	
mov bh,[62]	
sub dx,dx	
int 10	
ret	burada Enter yapıp kod yazmaktan çıkın
-n mavi.com	programın ismi belirleniyor (n - name)
-r cx	
30	30h = 48 Byte yer kaplayacağı belirtilir
-w	Sabit diske kayıt (w - write)
-u 100	unassemble komutu ile kodlarımızı inceliyoruz
18A2:0100 B412	MOV AH,12
18A2:0102 B310	MOV BL,10
18A2:0104 CD10	INT 10
18A2:0106 B84000	MOV AX,0040
18A2:0109 8ED8	MOV DS,AX
...	
-q	debug'dan çıkış (q - quit)

48 byte boyundaki mavi.com dosyamızı bir hex – editör ile açarsak, bizim doğrudan bir şey anlamadığımız şu yazıları görebilirsiniz:

```
00000 B4 12 B3 10 | CD 10 B8 40 | 00 8E D8 B8 | 00 06 B7 1F | □ □ □ □ © @ . Ä Æ □ À
00010 29 C9 B6 18 | 80 FB 10 74 | 04 8A 36 84 | 00 8A 16 4A | ) Â ° Ç ' □ t ° è ä ë □ J
```

00020 00 FE CA CD | 10 B4 02 8A | 3E 62 00 29 | D2 CD 10 C3 . _ -- □ | □ è > b.) Ê +

Her komutun bir mikroişlemci kod çevrimi vardır. e yani enter komutu ile makine komutu, yazı veya sayı girişi yapabiliriz. Sadece bu kodları kullanarak da program yazabiliriz. **MOV AH,12** yazmak yerine şunu da deneyebilirsiniz:

-e 100 b4 12 100 adresinden itibaren kod girişi (e - enter)
 -e 200 "Hafızaya bu yazılır\$" 200 adresine bilgi yazar, \$ string sonu demektir
 -e eff 11 10 1c 33 45 eff adresinden itibaren beş sayı yazdık

Başka bir zaman tekrar debug'da yazdığımız bu programı yüklemek istersek L yani load komutunu verebiliriz:

-n mavi.com dosya adını belirtiyoruz
 -l yükleme komutu (l - load)
 -u 100
 18CA:0100 B412 MOV AH,12
 18CA:0102 B310 MOV BL,10
 18CA:0104 CD10 INT 10
 18CA:0106 B84000 MOV AX,0040
 ...

Tüm sayılar tamsayıdır, negatif ve ondalıklı yani virgüllü rakamlar bayraklar ve diğer yardımcı yazmaçlarla kullanılıyor. FPU - Floating Point Unit denilen bölüm ile kayar noktalı işlemler daha hızlı hesaplanır. İşlemci yazmaçlarından olan bit boyutundaki sekiz adet bayrak ile işaret (negatif veya pozitif sayı) sign - SF, sıfır zero - ZF, elde carry - CF, eşlik parity - PF, yön direction - DF, taşma overflow - OF ile bölme, çarpma gibi işlemler yapılır.

-r f flag yazmaçlarını göster ve değiştir
 NV UP EI PL NZ NA PO NC - komut bekler, Enter ile geçebilirsiniz

$$(11010.10110)_2 = 2^4 + 2^3 + 2^1 + 2^{-1} + 2^{-3} + 2^{-4} = (26.6875)_{10}$$

$$(11010.10110)_2 = (1.101010110 * 2^5)_2 \text{ olarak da kullanılabilir}$$

$$(0,25)_{10} = 0.10^0 + 2.10^{-1} + 2.10^{-2}$$

$$(A2.B)_{16} = A.16^1 + 2.16^0 + B.16^{-1} = 162.687$$

On tabanındaki kayar noktalı bir sayı herhangi bir sayı sistemine çevrilirken; tamsayı kısmı 0 çıkana dek o sayı sistemine bölünüp kalanlar tersten yazılır; kesirli kısım ise yine 0 çıkana dek o sayı sistemi ile çarpılarak tamsayı kısmı yanyana yazılır:

$$(72.3)_{10} = (1001000.0100110)_2$$

72/2	Kalan = 0	.3*2=.6	Tamsayı=0
36/2	Kalan = 0	.6*2=.2	Tamsayı=1
18/2	Kalan = 0	.2*2=.4	Tamsayı=0
9/2	Kalan = 1	.4*2=.8	Tamsayı=0
4/2	Kalan = 0	.8*2=.6	Tamsayı=1
2/2	Kalan = 0	.6*2=.2	Tamsayı=1
1/2	Kalan = 1	.2*2=.4	Tamsayı=0
0			

"-" işareti debug'ın komut istemidir. Burada a, n, r gibi tek harfli komutlar kullanıyoruz. a 100 ile 100 ncü adresten itibaren yazmaya başlarız. Buradaki 100 prensip olarak kullanılan bir değerdir. Her program satırından sonra 100 yada diğer adres artı komut boyutu kadar adresin ilerlediğini görebiliriz. Komutların hafızada kapladığı alan kadar onların işlemci içinde ne kadar zaman harcadığı da önemlidir. Mesela div - bölme komutu 80486 için 24 saat darbesi ile biterken, benzer görevi yapan shr - sağa kaydırma sadece 3 saat çevrimine ihtiyaç duyar. İkiye bölmek veya çarpmak için en hızlı ve pratik komutlar shl çarpma ve shr bölmedir. Fakat kaydırırken taşan bitler yok olur.

mov ax, 8
 shl ax, 1 // sola bir bit kaydır, ax artık 0010h olur 10h = 16, iki ile çarpmak gibi
 ...
 mov ax, 4
 shr ax, 1 // sağa bir bit kaydır, ax artık 0002h olur, ikiye bölmek gibi

Bu işlemleri mul ve div ile şöyle yapabiliriz:

mov ax, 8

```

mov    bx, 2
mul    bx           //ax = ax * bx sonuç = 0010h = 16 (üst byte dx'te saklanır)
...
mov    ax, 4
mov    bx, 2
div    bx           //ax = ax / bx sonuç = 0002h (kalan dx'te saklanır)

```

Eğer bir byte bir byte ile çarpılırsa iki byte, iki byte iki byte ile çarpılırsa da dört byte alan kaplayan sayı oluşur. Bu sebeple eğer mul bl denilirse ax = al * bl olur, mul bx denilirse dx:ax = ax * bx olur. Bölmede de div bl gibi bir byte bölme yapılırsa (al = ax / bl) sonuç al'de kalan ah'de olur. div bx denilirse de (ax = dx:ax / bx) sonuç ax'de, kalan da dx'de saklanır.

```

-a 100
18A2:0100    mov ax,0011
18A2:0103    mov bx,2
18A2:0106    div bl
18A2:0108
-t 3

```

```

AX=0011 BX=0002 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000 ;17 / 2
DS=18A2 ES=18A2 SS=18A2 CS=18A2 IP=0103 NV UP EI PL NZ NA PO NC
18A2:0103 BB0200    MOV    BX,0002

```

```

AX=0011 BX=0002 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=18A2 ES=18A2 SS=18A2 CS=18A2 IP=0106 NV UP EI PL NZ NA PO NC
18A2:0106 F6F3    DIV    BL

```

```

AX=0108 BX=0002 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000 ;Kalan 01, Sonuç 08
DS=18A2 ES=18A2 SS=18A2 CS=18A2 IP=0108 NV UP EI PL NZ NA PO NC
18A2:0108 0000    ADD    [BX+SI],AL    DS:0002=00

```

```

-a 100
18A2:0100    mov ax,0100           ; 256 / 5 = 51, kalan 1 olur
18A2:0103    mov bx,05
18A2:0106    div bx
18A2:0108
-r ip           ; program 100'den itibaren çalışması gerekir
IP 0108
:0100
-t3

```

```

AX=0100 BX=0002 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=18A2 ES=18A2 SS=18A2 CS=18A2 IP=0103 NV UP EI PL NZ NA PO NC
18A2:0103 BB0500    MOV    BX,0005

```

```

AX=0100 BX=0005 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=18A2 ES=18A2 SS=18A2 CS=18A2 IP=0106 NV UP EI PL NZ NA PO NC
18A2:0106 F7F3    DIV    BX

```

```

AX=0033 BX=0005 CX=0000 DX=0001 SP=FFEE BP=0000 SI=0000 DI=0000 ; ax = 51(onluk sistemde), dx = 1
DS=18A2 ES=18A2 SS=18A2 CS=18A2 IP=0108 NV UP EI PL NZ NA PO NC
18A2:0108 0000    ADD    [BX+SI],AL    DS:0005=9A

```

Örneğin **MOV AH,12** komutu 2 byte yer kaplayarak kod hafızasına b412 yazılmasını sağladı. Tabii ki buradaki 12 sayısı bildiğimiz onluk sistemdeki 12 değil. *Heksadesimal* olarak düşünmeliyiz. 0 ile 9 arası ve A 10, B 11, C 12, D 13, E 14, F de 15 rakamı olarak kullanılır.

$$\begin{aligned}
 12h &= 16^1 \cdot 1 + 16^0 \cdot 2 = 18 \\
 FFh &= 16^1 \cdot 15 + 16^0 \cdot 15 = 255 \\
 0Ch &= 16^1 \cdot 0 + 16^0 \cdot 12 = 12 \\
 FF00h &= 16^3 \cdot 15 + 16^2 \cdot 15 + 16^1 \cdot 0 + 16^0 \cdot 0 = 65280
 \end{aligned}$$

İkilik sistemden de birkaç örnek yapalım, genellikle yazımda dörder dörder ayrılırlar:

$$\begin{aligned}
 1100 &= 0Ch = 2^3 \cdot 1 + 2^2 \cdot 1 + 2^1 \cdot 0 + 2^0 \cdot 0 = 12 \\
 1111\ 0001 &= F1h = 2^7 \cdot 1 + 2^6 \cdot 1 + 2^5 \cdot 1 + 2^4 \cdot 1 + 2^3 \cdot 0 + 2^2 \cdot 0 + 2^1 \cdot 0 + 2^0 \cdot 1 = 241
 \end{aligned}$$

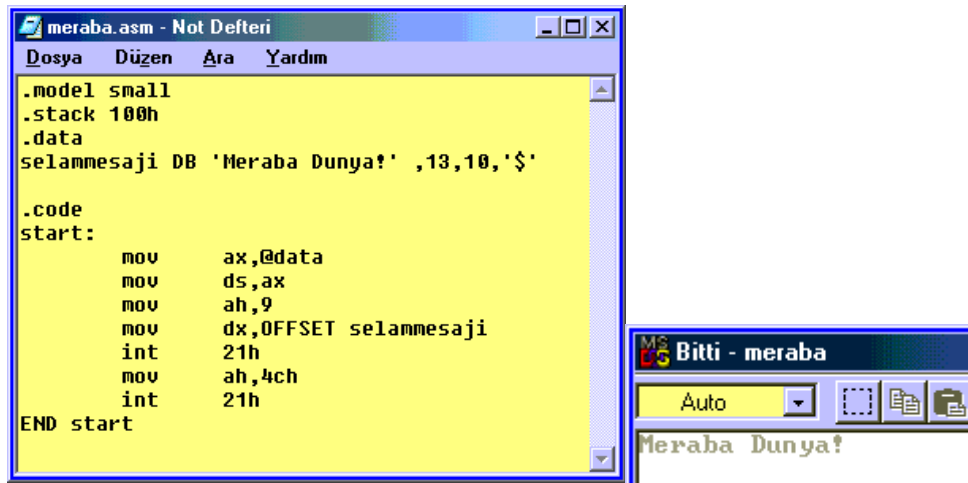
Eminim 128, 256, 1.44, 65535, 1024 gibi rakamları bir yerlerde görmüşsünüzdür. Örneğin RAM'lerimiz 1024'ün katları şeklindedir. 16 MB, 32 MB, 48 MB gibi... Eğer 128 MB RAM'imiz var ise bilgisayarın POST – Power On Self Test ilk açılışında 131.072 KB gibi bir rakam görürsünüz. Nedeni megabyte ve kilobyte çevriminden dolayıdır. Anakart veya bir ekran kartının kitapçığını incelemenizi tavsiye ederim. Kullanılan terimler, deyimler ve kısaltmaların anlamlarına bakınız. (B byte, K kilo, M mega demektir). Binin katları şeklinde olması için en yakın sayı 2^{10} yani 1024'dür.

$$\begin{array}{ll} 1 \text{ B} = 8 \text{ bit} & 1024 \text{ B} = 2^{10} \text{ B} = 1 \text{ KB} \\ 1024 \text{ KB} = 1 \text{ MB} & 131.072 \text{ KB} = 128 \text{ MB} \end{array}$$

Temel birimler ise; tera T 10^{12} , giga G 10^9 , mega M 10^6 , kilo K 10^3 , hecto h 10^2 , deci d 10, centi c 1, mili m 10^{-3} , micro μ 10^{-6} , nano n 10^{-9} , pico p 10^{-12} diyebiliriz. Fakat gün geçtikçe daha büyük ve daha küçük birimlere ihtiyacımız arttığı için değişik birimlerle karşılaşacağız. Mesela perabyte PB 10^{15} , eksabyte EB 10^{18} gibi...

2. meraba.asm (merhaba dünya!)

(Ornekler\Asm\asm.zip*meraba.asm)



Resim 4 – Assembler selamı

Bir metin editörü ile yazdığınız kaynak dosya, Tasm – Turbo Assembler veya Masm – Microsoft Assembler ile *.obj dosyası oluşturulur. Eğer hata kalmamış ise, Tlink veya Link ile *.exe veya *.com dosya haline gelir. *.com dosya daha küçüktür çünkü sadece MS-DOS geleneksel hafıza bölgesi olan 1 MB içine yerleşebilir, boyut olarak 64 kilobyte'ı geçmezler; *.exe dosya ise RAM'in herhangi bir yerine yerleşir ve büyüklükleri mega byte'larca olabilir. DOS – Disk Operating System komut isteminde mem komutunu vererek, *.com dosyalar için en fazla ayrılabilen boş alanı görebilirsiniz. Şu kodları not defteri ile "meraba.asm" olarak yazıp tasm ile derleyebilirsiniz:

```
.model small
.stack 100h
.data
selammesaji DB 'Meraba Dünya!' ,13,10,'$'      ;'$' metin dizi sonu belirtici

.code
start:
    mov     ax,@data
    mov     ds,ax
    mov     ah,9                                ;DOS yazma fonksiyonu
    mov     dx,OFFSET selammesaji              ;veriyi işaretle
    int     21h                                ;yaz
    mov     ah,4ch                              ;DOS program sonu
    int     21h                                ;program sonu
END start
```

meraba.exe dosya oluşturmak için, MS-DOS Komut isteminde şunları yapabilirsiniz. Eğer Tasm hata mesajları veriyorsa bunları düzeltmeden meraba.obj oluşamaz ve Tlink işe yaramaz.

```
c:\>tasm meraba.asm
Turbo Assembler Version 3.2i Copyright (c) 1988, 1992 Borland International
Serial No: Tester:
```

```
Assembling file: meraba.asm
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 393k
```

```
c:\>tlink meraba.obj
Turbo Link Version 5.1 Copyright (c) 1992 Borland International
```

```
c:\>_
```

Artık "meraba" yazıp Enter tuşuna basınca ekrana gelen "Meraba Dünya!" yazısını görebilirsiniz. meraba.obj 260 byte, meraba.map 232 byte, meraba.asm 209 byte ve meraba.exe 544 byte olmalı.

'Meraba Dünya!' yazısının yanındaki 13 ve 10 Enter ve satır besleme – Line Feed tuşlarının taklididir. CR – Carriage Return şaryo / taşıma dönüşü ASCII* kodu 13, LF – Line Feed satır besle ASCII kodu ise 10'dur.

Birçok int komutundan önceki ax veya ah yazmacına atılan değerlere göre kesme işlemi uygulanır. Üstteki programda iki adet int 21h komutu var, ama etkileri birbirinden çok farklı. Doğrudan int 21h komutu verilseydi istediğimizi elde edemezdik.

int 09h klavye, int 10h ekran BIOS içinde tanımlanmıştır, int 33h fare de mouse.sys ile belleğe yerleşir. Bu interrupt – kesmelerle ilgili örneklere kitabın ilerleyen kısımlarda bulabilirsiniz.

256 adet 0-FFh arası yazılım kesmesi olabilir. Hafızanın ilk alanı interrupt bilgilerini taşır ve her biri dört byte dallanılacak adres bilgisi olan yer kaplar. Bir kesmenin alt fonksiyonları AH yazmacı ile belirtilir. AH sekiz bit olduğu için en fazla 256 fonksiyon tanımlanabilir. Mikroişlemci interrupt komutu aldığı anda, o anki yazmaç bilgilerini ve program adresini saklar ve kesmeyi gerçekleştirir. Kesme işlemi bitince ana program kaldığı yerden çalışmaya devam eder. Donanım kesmeleri de 15 adettir. Klavye, fare gibi donanımlardan gelecek tetikleme ile çalışırlar. İşletim sisteminin sifıra bölüm gibi hatalara karşı kullandığı birçok internal – iç kesmeleri de bulunur.

"Ornekler\Asm\asm.zip*user_interrupt.txt" örneğinde int 60'ı kendimiz programlayıp, belli bir hafıza bölgesine dallanmayı debug programında şöyle sağlıyoruz:

60h x 4 = 180h adresi int 60'ın başlangıç adresidir. Aşağıdaki 00 00 00 00 bilgisi ile bu kesmenin boş olduğunu anlıyoruz.

-d 0:180 L4	Length 4 byte göster (d - dump)
0000:0180 00 00 00 00	
-e 0:180 00 01 6d 18	186d:0100 adresine yönlendireceğiz
-e 186d:0200 "Tarık BAĞRIYANIK\$"	0200 adresinde bilgi var, \$'a dikkat, string sonu
-a 186d:0100	bu adrese de kendi programımızı yazıyoruz
186D:0100 mov ax,186d	offsetimiz 186d
186D:0103 mov ds,ax	mov ds,186d hatalı kullanımdır
186D:0105 mov dx,0200	0200 adresindekileri ekrana yaz
186D:0108 mov ah,09	
186D:010A int 21	
186D:010C ret	interrupt bitirme komutu (iret komutu da olabilirdi)
186D:010D	Enter ile programı bitirin
-a 186d:1000	int 60'ı çağıran başka bir program
186D:1000 int 60	
186D:1002 mov ah,4c	programı bitirme
186D:1004 int 21	
186D:1006	
-g =186d:1000	programı çalıştır (g - go)
Tarık BAĞRIYANIK	ekrana bilgi yazıldı...

Aslında interrupt diye yaptığımız bu örnek **call** komutu ile aynı işi yapıyor. **call** ile alt programları çağırabilir **ret** komutu ile ana programa dönülebilir.

Assembler dilini donanıma ulaşmak için bize yardımcı olan dil olarak görebiliriz. Bu sebeple bilgisayarımızı kilitleme ihtimalimiz bir hayli fazla, dikkatli yazınız. Dediğim gibi tüm satırları tek tek açıklamak çok zor, şimdilik bu bunların op-code'lar olduğu, bir şeylerin kısaltması olduğunu bilin, yeterli. Mesela mov, move yani sağdaki değeri al, soldakine yapıştır demek. Yapılan işlemler sanki matematiksel yani topla, çıkar, ekle, eksilt gibi değil mi? Sonuçta hafıza veya yazmaçlarla ilgili işlem yaparken hesap yapmayı da öğrenmemiz gerekiyor.

Eğer assembler'da programlama yapmaya karar verdiyseniz on, on beş komutu iyi bilmeniz gerekiyor. CD'deki "CD2\Programlama\HelpPC\HelpPC.com" komutlar ve yazmaçlar hakkında ayrıntılı bilgiler veriyor.

* Ek 3 – ASCII Kod Tablosuna bakınız.

3. kucuk.asm (büyük harfe çevirir)

(Ornekler\Asm\Diger.zip*KUCUK.ASM)

Cevrilecek cumle <10 karakter>:bağrıyanı
CEv_ BAĞRıYANı

Resim 5 – 10 karakteri büyük harfe çeviren program (Türkçe karakterlere dikkat)

```
.MODEL SMALL
.STACK 100H
.DATA
    buffer label byte
    max_uzun db ?
    katar db 10 dup(?)
    girilen db ?
    mesaj db "Cevrilecek cumle (10 karakter):$"
    crlf db 10,13,'$'

.code
basla:
    mov ax,@data
    mov ds,ax
    mov dx,offset mesaj          ; mesaj değişkeni ekrana basılacak
    mov ah,09h
    int 21h
    mov dx,offset buffer
    mov buffer,10
    mov ah,0ah
    int 21h
    xor si,si
    xor cx,cx
    mov cl,girilen
    cmp cl,0
    je son

buyuk:
    cmp katar[si],'a'
    jl devam
    cmp katar[si],'z'
    jg devam
    sub katar[si],20h

devam:
    inc si
    loop buyuk
    mov katar[si+1],'$'

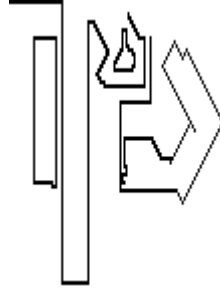
goster:
    mov dx,offset crlf
    mov ah,09h
    int 21h
    mov dx,offset katar
    mov ah,09h
    int 21h

son:
    mov ax,4c00h
    int 21h
end basla
```

kucuk.exe dosyası çalıştığında küçük harfli on karakter girin ve Enter'a basın. On harfli olmasına dikkat edin. Programı değiştirebilmek için edit veya not defteri ile kucuk.asm dosyasını açabilirsiniz. İlk kez not defterinde aşağıdaki programı yazıp kaydederken de çift tırnak içinde "dosyaadi.asm" şeklinde yazıp kaydedin, böylece sonu *.txt kalmaz.

4. cizim.asm (klavye ile çizim yapma)

(Ornekler\Asm\Diger.zip*CIZIM.ASM)



Resim 6 – Grafik ekranda çizim yapmak.

Klavyeden yön tuşları, Home, End, Page Up, Page Down ve Insert tuşları ile çizim yapılıyor. Insert ile çizim yapma ve yapmama ayarlanıyor. Belki siz de yaptığınızı kaydetmeyi eklemek istersiniz (tamam Print Screen tuşu da işi görüyor zaten)... Bu örneği de önceki örnekteki gibi derleyebilirsiniz. Ekranın dışına da çıkmaya çalışmayın, garip hatalar göreceksiniz.

kes.asm dosyası da buna benzer bir örnek, renk seçimi dahil edilmiş. Hatta bilgisayarınızı bile kilitleyen kayıt bile var. Biraz inceleyin, belki sizin hoşunuza gider.

↑↔RENK SEÇİM <ESC> RENK KABUL EDİLDİ

Resim 7 – kes.asm'nin ekran görüntüsü

<pre>.MODEL SMALL .code org 100H basla: jmp ilk instusu db 0h deltusu db 0h ilk: mov ah,00 mov al,06 int 10h xor cx,cx xor dx,dx tusall: xor ax,ax int 16h cmp al,1bh jne ara mov ah,0 mov al,03 int 10h mov ax,4c00h int 21h ara: cmp ah,47h jl tusall cmp ah,53h jg tusall cmp ah,52h je ins_t</pre>	<pre> cmp ah,53h je del_t cmp ah,47h je home cmp ah,48h je yukari cmp ah,49h je pgup cmp ah,4bh je sol cmp ah,4dh je sag cmp ah,4fh je endt cmp ah,50h je asagi cmp ah,51h je pgdn jmp tusall ins_t: not instusu jmp tusall</pre>
--	---

```

del_t:    not deltusu
          jmp tusal
home:     call noktanokta
          dec cx
          dec dx
          jmp tusal
yukari:   call noktanokta
          dec dx
          jmp tusal
asagi:    call noktanokta
          inc dx
          jmp tusal
sag:      call noktanokta
          inc cx
          jmp tusal
sol:      call noktanokta
          dec cx
          jmp tusal
endt:     call noktanokta
          dec cx
          inc dx
          jmp tusal
pgup:     call noktanokta
          inc cx
          dec dx
          jmp tusal
pgdn:     call noktanokta
          inc cx
          inc dx
          jmp tusal

noktanokta proc
          cmp instusu,0h
          jne don
          call noktakoy

```

```

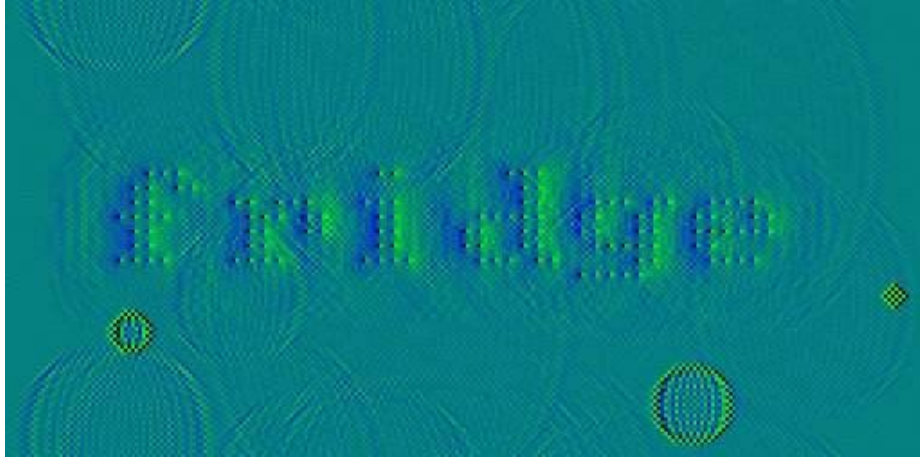
don:      ret
          noktanokta endp
          noktakoy proc near
            push es
            push bx
            push cx
            push dx
            push di
            xor di,di
            mov bx,0b800h
            mov es,bx
            MOV di,cx
            shr di,1  L harfi değil, bir yazılır
            shr di,1
            shr di,1
            mov ax,dx
            shr ax,1
            mov bl,80
            mul bl
            add di,ax
            and dx,1
            push cx
            mov cl,13
            shl dx,cl
            pop cx
            add di,dx
            and cx,7
            mov bl,7
            sub bl,cl
            mov cl,bl
            mov ah,1
            shl ah,cl
            mov al,es:[di]
            or al,ah
            mov es:[di],al
            pop di
            pop dx
            pop cx
            pop bx
            pop es
            ret
          noktakoy endp
          end basla

```

5. Fridge (intro yapımı)

(Ornekler\Asm\Fridge.zip\Fridge.asm)

Assembler dilinde yapılmış hoş bir grafik efekti programı. Gerçekten kaliteli bir çalışma. *.com olarak derlemeniz gerekiyor. İngilizce açıklamaları çözebilirseniz siz de güzel değişiklikler yapabilirsiniz. Siz de "buzdolabını kapat" lafını rahatlıkla değiştirebilirsiniz:



Resim 8 – Fridge demosu

```

IDEAL
P386
MODEL tiny
CODESEG
ORG 100h

Start:
  mov al,13h ; assume ah=0
  int 10h ; set mode 13h

  mov ah,09h
  mov dx,OFFSET Msg
  int 21h

  mov ax,cs
  add ah,10h ; ie. add cs,1000h
  mov ds,ax ; setup buffer segment 1
  add ah,10h ; ie. add cs,2000h
  mov es,ax ; setup buffer segment 2

  push 0A000h ; vga-mem ptr
  pop fs

;----- clear water buffers -----
  xor ax,ax
  xor cx,cx
@@c: mov [di],al
      stosb
      loop @@c ; clear ds and es buffers

;---- Set Palette -----
  mov dx,03C8h ; assume ax = 0
  out dx,al
  inc dx

@@p:
  xor al,al
  out dx,al ; red = 0
  mov al,ah
  shr al,2

```

```

out    dx,al      ; green = ah/4 (0->63)
neg     al
inc     al
out    dx,al      ; blue = -ah/4+1 (64->0)
inc     ah
jnz    SHORT @@p

pusha      ; to decrease sp, and to store cx=0

;----- Main loop begins -----
Main:
;----- calculate water -----
xor     di,di
mov     cx,32768
@@w:    mov     ax,[di-512]
add     ax,[di+512]
add     ax,[di-2]
add     ax,[di+2]
sar     ax,1      ; /2

sub     ax,[es:di]
mov     dx,ax
sar     dx,5
sub     ax,dx
stosw   ; mov [es:di],ax    add di,2
loop    @@w

;----- vertical retrace ----
mov     dx,03DAh
@@vr1:  in      al,dx
test    al,8
jz      SHORT @@vr1

;----- change water buffers -----
push    ds
push    es
pop     ds
pop     es      ; exchange ds/es

;----- paint water -----
xor     si,si
mov     di,02D20h ; y=36, x=32
@@o:    inc     ch      ; assume cx=0, inc ch -> cx=256
@@i:    mov     dx,128
lodsw   ; mov ax,[si]    add si,2
sub     ax,[si]      ; x+1, ([si+510] for y+1)
sub     dx,ax        ; color = water[y][x]-water[y][x+1](should be y+1)

cmp     dx,0
jnl     SHORT @@d1
mov     dl,0
@@d1:   cmp     dx,255
jng     SHORT @@d2
mov     dl,255
@@d2:

mov     [fs:di],dl   ; slow putpixel
inc     di
loop    @@i

add     di,64
cmp     si,0
jne     SHORT @@o

;----- add drops -----
popa      ; pop regs from the stack

```

```

add ax,26500 ; .
rol ax,1 ; fake random generator
inc cx ; frame counter
pusha ; push changed regs to the stack

test cl,00000011b ; only 1 drops / 4 frames
jnz SHORT @@nd
mov di,ax
and al,00000011b ; drop height (0-7)
add [di],al
@@nd:
;----- add text -----
mov al,10
and ch,00000011b
mul ch
mov bh,al
push bx
cmp cl,100
ja SHORT @@skip

mov di,24641 ;(y=48,x=32)*2+1(1=for offset to second byte)
@@to: mov cx,64 ; x-pixels of text
@@t: mov al,[fs:bx]
cmp al,0
jz SHORT @@no
inc [BYTE es:di] ; add [WORD es:di],256 (256=height)
@@no:
inc bx
add di,8

loop @@t
inc bh ; add bx,(320-64)
add di,(256*3*2)
cmp di,41025 ; (y=48+32, x=32)*2 + 1
jb SHORT @@to
@@skip:
pop bx

;----- key check -----
in al,60h ; read scancode
cbw ; clear ah, for later use
dec ax
jnz Main ; jmp if Esc not pressed
;----- Main loop Ends -----
mov al,03h ; ah assumed to be 0
int 10h ; set mode 03h
int 20h ; terminate program, safer than "ret" in win95

Msg DB "Close",13,10,"tha",13,10,"fridge",13,10,"!","$" ;ascii-$ string
;Msg DB "Be",13,10,"HaPpY",13,10,"dont",13,10,"WorrY","$" ; alternative
END Start

```

Mikroişlemci ile ilgili notlar şimdilik bu kadar.

Bölüm 2 – Basic

Basic (Beginner's All Purpose Symbolic Instruction Code) dili ile oldukça uğraşmama rağmen burada pek yer vermeyeceğim. Pascal veya C dilini görünce insan Basic'i adı gibi basit ve kaba buluyor. Tamam kızmayın Basic'ciler. Microsoft her işi kolaylaştıracağı derken kullanıcının kendilerine özgü tasarım yapmalarına engel oluyorlar ve sanki at gözlüğü takmış gibi gitmemizi istiyorlar.

1. piyano.bas (müzik)

(Ornekler\VB\EskiBasic\GWBASIC.zip*piyano.bas)

```
GW-BASIC 3.20
(C) Copyright Microsoft 1983,1984,1985,1986
BASIC Interpreter Rev. 2.0
58476 Bytes free
Ok
load "piyano
Ok
list
10 COLOR 1,2:CLS
20 LOCATE 12,20:PRINT "BU BİR PİYANO PROGRAMIDIR"
30 T=RND*10000
40 SOUND T,5
50 FOR I=1 TO 10000
60 NEXT
70 CLS
80 LOCATE 12,20:INPUT "NOTANIZ(CDEFGA-F+ D+ A+ ...)===== ";A$
90 PLAY A$
100 GOTO 80
Ok
run
```

1LIST 2RUN- 3LOAD" 4SAVE" 5CONT- 6,"LPT1 7TRON- 8TROFF- 9KEY 0SCREEN

Resim 1 – GWBasic ekranı (yıl 1992 gibi...)

Bir metin – yazı girişi ile nota çalınıyor. GWBasic ile yazdım. Resimde görüldüğü gibi Basic Interpreter yazıyor. Yani programlarımızı *.exe yapamıyorduk o zamanlar.

```
10 COLOR 1,2:CLS
20 LOCATE 12,20:PRINT "BU BİR PİYANO PROGRAMIDIR"
30 T=RND*10000
40 SOUND T,5
50 FOR I=1 TO 10000
60 NEXT
70 CLS
80 LOCATE 12,20:INPUT "NOTANIZ(CDEFGA-F+ D+ A+ ...)===== ";A$
90 PLAY A$
100 GOTO 80
```

GWBasic 3.2'nin kullanımı kolay. F1 programın kod listesini paldır küldür ekrana verir, eğer program uzunsa yani. F2 çalıştırma, F3 dosya yükleme. Eğer dosya adını hatırlamıyorsanız "files" komutu ile dosyaları görebilirsiniz. Ekranda renkler birbirine girerse F10 ile varsayılan siyah beyaz ekrana geri dönebilirsiniz. "cls" ekranı siler, "system" GWBasic'i kapatır, program kilitlenirse Ctrl+Pause ile programı kırabilirsiniz.

Mikroişlemci komutları gibi satır numaraları ile program çalışıyor. Yani Goto komutu ile istediğiniz an istediğiniz yere atlayabiliyorsunuz. Ne güzel değil mi? Bu arada sanatmuz.bas bir Türk müzikseverin yaptığı GWBasic ile program, "Ornekler\VB\EskiBasic\QBasic.zip*Examples" dizininde (Qbasic 4.5 ile rahat çalışıyor). Muhakkak inceleyin, güzel müzikler var.

2. kutu.bas (kutu şekli)

(Ornekler\VB\EskiBasic\QBASIC.zip*EXAMPLES\kutu.bas)

*.bas dosyaları not defteri veya edit'e atmayı denediniz mi? Diğer diller *.c, *.cpp, *.asm, *.pas, *.htm, *.asp dosyaları da dışarıdan açılıp edit edebilirsiniz. Ctrl+X, Ctrl+C gibi kısayollara alışanlar, Ctrl+Ins, Shift+Ins gibi tuşlara alışamayanlar bu şekilde program yazabilir.

```
DECLARE SUB KUTU (A!, B!, C!, D!)
```

```
CLS
```

```
KUTU 4, 5, 20, 70
```

```
LOCATE 4, 5: PRINT "X"
```

```
LOCATE 20, 70: PRINT "X"
```

```
SUB KUTU (A, B, C, D)
```

```
ORTX = (A + C) \ 2
```

```
ORTY = (B + D) \ 2
```

```
XX1 = ORTX
```

```
XX2 = ORTX + 2
```

```
YY1 = ORTY
```

```
YY2 = ORTY + 2
```

```
FOR XX2 = ORTX TO C - 1
```

```
XX1 = XX2 - 1: IF XX1 < A THEN XX1 = A
```

```
'-----
```

```
LOCATE XX1, YY1: PRINT " "; STRING$(YY2 - YY1 - 1, "="); " "
```

```
FOR SUTUN = XX1 + 1 TO XX2 - 1: LOCATE SUTUN, YY1: PRINT " "; STRING$(YY2 - YY1 - 1, " "); " ": NEXT
```

```
LOCATE XX2, YY1: PRINT "L"; STRING$(YY2 - YY1 - 1, "="); "L"
```

```
'-----
```

```
FOR I = 1 TO 7000: NEXT
```

```
NEXT
```

```
FOR YY2 = ORTY TO D
```

```
YY1 = YY2 - 1: IF YY1 < B THEN YY1 = B
```

```
'-----
```

```
LOCATE XX1, YY1: PRINT " "; STRING$(YY2 - YY1 - 1, "="); " "
```

```
FOR SUTUN = XX1 + 1 TO XX2 - 1: LOCATE SUTUN, YY1: PRINT " "; STRING$(YY2 - YY1 - 1, " "); " ": NEXT
```

```
LOCATE XX2, YY1: PRINT "L"; STRING$(YY2 - YY1 - 1, "="); "L"
```

```
'-----
```

```
FOR I = 1 TO 7000: NEXT
```

```
NEXT
```

```
END SUB
```

Aslında bu programın pek bir özelliği yok fakat artık burada ilk kez karşılaştığımız "Sub" kavramı ortaya çıktı. F2 ile program içindeki alt programlara - **subroutine** ulaşabilirsiniz. Ana modülde (programda)

Sub deneme (degisken)

yazdığınız anda kendinizi ayrı bir dünyada ana program dışında program yazmaya devam ettiğimizi görürsünüz. Bu insanda bağımlılık yapıyor, çünkü diğer DOS programlama dillerinde tüm kod aynı yerde oluyor. Burada ise ayrı pencerede prosedür veya fonksiyon oluşturuyorsunuz.

Prosedür içine veri yollanabilir fakat prosedür değer döndürmez. Oysa fonksiyon içine değer alabilir fakat her zaman kendi adı üzerine değer dönderir. Dönen fonksiyon sonucunun değişken tipi metin, tamsayı veya kayar noktalı olduğu muhakkak belirtilir. Bir alt programın prosedür mü yoksa fonksiyon olarak mı yazılacağına böyle karar veriyoruz.

```
FUNCTION kup (x AS INTEGER)
```

```
    kup = x * x * x
```

```
END FUNCTION
```

"Run*Make EXE File..."¹ komutu da diğer dillerde yok. Oysa C ve Pascal'da bunlar anında oluşturuluyor ve programcı ayrıca *.exe (executable – çalıştırılabilir) hale getirmeye uğraşmıyor. VB'nin de böyle bir huyu var. Eğer ilk kez kendi programınızı yazıyorsanız, bir satırı yarım bırakıp yada bilmeden hatalı yazarsanız hemen hatanın olduğu yeri belirten mesajla karşılaşsınız. Yani anında Basic hata denetimi yapıyor. Bilmem ne kadar rahat...

Tüm bunların sebebi Basic dilinin aslında interpreter – yorumlayıcı, diğer dillerin compiler – derleyici olarak çalışmasındandır. Yorumlamak demek tüm programı makine diline çevirmek yerine sadece *çalıştırmak*, işlemek demektir. Ne gerek varsa... Derleyiciler ise kaynak programı makine diline çevirerek (önce *.obj sonra hata kalmayınca *.exe) çalıştırırlar. *.obj dosyaların bir faydası da birden fazla programın birleştirilerek tek *.exe oluşmasına izin vermeleridir. Ana programınız alt programlardan oluşuyorsa TP (uses komutu ile) ve TC (#include komutu ile) dilleri zaten onları birleştiriyor. Makine dili de zaten şu bildiğimiz *.exe ve *.com uygulama dosyalarıdır. Tabii ki o kadar bir ve sıfırın hangisi kod, hangisi rakam ve hangisi yazı olduğunu ancak derleyici programlar halledebilir.

Makine dili düşük seviyeli, assembly dilleri orta düzey ve Basic, C, Pascal gibi diller de yüksek seviyeli olarak kabul ediliyor. Yüksek seviyede komutlar konuşma diline yakındır. Yüksek seviye dillerde programın ilk yazıldığı haline kaynak program (source), makine diline çevrilmiş haline de amaç program (object) adı verilir. Basic hem compiler olarak yapabilir amaç programa çevirebilir, hem de derlemeden interpreter olarak çalıştırabilir. Interpreter halinde programı çalıştırırken ilk karşılaştığı hatada durup, hata tipini belirtir. Compiler halinde tüm hatalar giderilmeden (mantık hataları hariç) çalıştırılmaz.

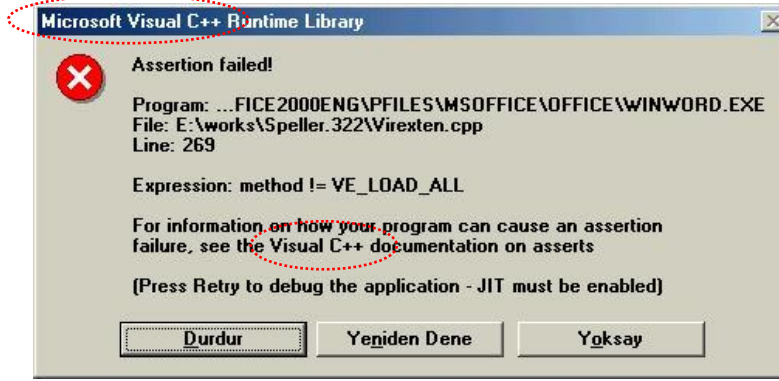
Dili tam bilmemek ilk hatalar iken daha ileri seviye programcılıkta mantık hataları daha çok göze çarpıyor. En iyi yöntem algoritma oluşturmak ve akış diyagramı ile onu desteklemek. Elinizde kalem kağıt olmadan karmaşık problemleri çözmek çok zordur. Takıldığınız noktaları kendi anladığınız yöntemlerle de şekillendirip planlamak çok rahatlatıcıdır. Genellikle ilham bilgisayar başında değilken daha rahat geliyor. Çünkü bilgisayar başında iken komut kargaşası, ilgiyi dağıtan başka olaylar programa konsantre olmayı engelliyor. Hataları gidermekle de iş bitmiyor; yeni istekler, güncellemeler programcıların daha çok zaman harcadıkları konulardır. Zor bir problemi gidermek, bazen kolay gözüken bir işlemde daha kısa sürebiliyor. Algoritmanın diğer faydaları da, daha az kodlama ile problem çözümü sunması, komut hatalarını en aza indirmesi ve daha kısa sürede program oluşturulmasıdır.

Öğrenmek istediğiniz dilde değişken tanımlama, (aritmetik) operatörler, karşılaştırmalar, sayaçlar, döngüler ve işlem öncelikleri konularını iyi anlamalısınız.

¹ QuickBasic 4.5 versiyonunda derleme imkanı var. Windows CD'sindeki "\\tools\\oldmsdos\\qbasic.exe" 1.1 versiyonu ve derleme imkanı yoktur. Sadece editör ve çalıştırıcı.

Bölüm 3 – C Dili

C ve C++ (*si plas plas diye okunur*) programcılığı temel programcılıktan bir seviye daha üsttedir. Assembler'dan sonra donanım en yakın dil sayılır. Öğrenilmesi en zor dillerdendir. İşlemci, hafıza gibi kaynakları istediğiniz gibi kullanabilirsiniz. Bu sebeple C dili kullanılarak 1978 yılında Unix işletim sistemleri üretilebilmiştir. MS-DOS'da yazdığınız kaynak kodunuzu alıp Unix'de de derleyebilirsiniz. Microsoft da Office uygulama paketinin ve Windows işletim sisteminin üretiminde C dili kullanmıştır.



Resim 1 – Office uygulaması MS-VC++ ile yapılmıştır

Assembler ve C programcılığında programcı ile donanım arasında bir kontrol birimi yoktur. Hataları programcının bilerek yaptığı kabul ediliyor. Bu sebeple en çok bilgisayarı kilitleyen programlama dilleri bunlardır. Tabii ki program durup dururken bilgisayarı kilitlemez yada bilgisayarın garip davranmasına sebep olmaz. Yazdığımız programdaki yazım hataları, dili tam bilmeme ve dikkatsizlik ana sebeplerdir. Girişte yazdığım böcek ayıklama ile ilgili yazı daha çok bu diller için geçerli. Dilin okunaklılığı da çok önemli. Eğer algoritmasız, karmakarışık kod yazarsak kendimiz bile işin içinden çıkamayız.

Hemen her dilin ayrılmaz parçası olan metin işlem, giriş / çıkış ve ekran kontrol komutları C dilinde gizli yerlerde. #include kısımlarına bir şey yazmazsanız ekrana basit bir uyarı bile çıkaramazsınız. Library yani kütüphaneler içindeki komut kümeleri öyle zengindir ki işlevsel olarak programınızda hangisini kullanacaksanız seçimde zorlanabilirsiniz. Mesela getchar() Enter tuşunu bekleyen ekrana gözüken tek tuş okuma, getch() Enter tuşu beklemeyen ekrana gözükmeyen tek tuş okuma, getche() Enter tuşu beklemeyen ekrana gözüken tek tuş okuma fonksiyondur. scanf(), gets() gibi diğer standart veri giriş komutları da vardır.

Çok gelişmiş program yapısına ve icra akış kontrol komutlarına sahiptir. Ayrıca operatör kümesi de (+, -, *, /, >, <, &, =, ++, %, !=, >>, ^, ->, ?:, sizeof() ...) geniştir. Üs almak için pow (taban, üs); kullanılabilir. Makine dili seviyesindeki tüm operasyonların C dilinde karşılığı bulunur. Yani sistem programcılığı ve benzeri ileri seviye programcılığına elverişlidir.

Ana program main() içinde olmalıdır. Eğer main() içinden çağrılmayan bir kod yazmışsanız asla o kısım çalıştırılmaz. Pascal dilinde de **end.** deyiminden sonra yazılan kısımlar sadece açıklama satırı gibi anlaşılır.

```
#include <stdio.h>
int main(int argsayi, char *args[ ])
{
    int k;
    for (k=0 ; k<argsayi; k++)
        printf ("arg %d = %s\n", k, args[k]);
    return 2001;
}
```

`int main (int argumansayisi, char *argumanlar [ ])` şeklinde aslında bir fonksiyon olarak da kullanılabilir. Yani programınız değer dönderen, komut isteminden değer okuyan bir yapıda olabilir. void – boş ifadesi ile main() veya başka fonksiyonlarımızın değer almayacağını veya değer döndürmeyeceğini bildiririz. Birden fazla main fonksiyonu tanımlanamaz. Pascal dilinde ise prosedürler değer döndürebilmesi için var deyimini kullanılır:

```
var x,y:integer;           (*global x ve y değişkeni*)
procedure swapint (var x:integer; var y:integer);
```

```

var z: integer;
begin
  z:=x; x:=y; y:=z;
end;
begin
x:=10; y:=20;
swapint(x,y);      (*10 ve 20 prosedür içine atılıp, x ve y'nin yeni değerleri gelir*)
writeln(x,y);      (*ekrana 2010 yazar, var denilmeseydi 1020 derdi...*)
end.

```

Tüm değişkenler kullanım alanlarının üstünde tanımlanmak zorundadırlar. Mesela tüm programda geçerli olacak değişkeni "main(){" deyiminden önce yazabilirsiniz. { } blokları içinde kullanılacak değişkenler de "{" işaretinden hemen sonra yazılır. Statik – durağan tanımlanan değişkenler data segment – veri bölgesi, dinamik değişkenler de stack segment – yığın bölgesi denen hafıza kısımlarında saklanır. Değer verilmemiş global değişken 0 – sıfır, lokal değişken ise hafızadaki belli olmayan bir değere eşittir. C++ dilinde :: operatörü ile bir üst bloktaki aynı isimdeki değişkene ulaşabilirsiniz.

Komutlar yazılırken küçük büyük karakter fark eder. (case sensitive – harf duyarlı) Örneklerdeki gibi aynen kullanmanız gerekir. VB dilinde DenemeDeger tanımlayıp sonra denemeDeger yazarsanız tüm değişken adlarını son değişikliğe göre güncellenir.

Bir şart eğer doğru ise 1, yanlış ise 0 değeri dönderir. Bu sebeple şart ifadeleri hesap işlemlerinde kullanılabilir.

```

kalan+=(notu<0.5);      //unary işlemler
anlamı notu 0.5'den küçük ise kalana 1 ekle.

```

0 – sıfır dışındaki tüm tam sayılar negatif de dahil TRUE (*tru diye okunur*) yani doğrudur, 0 da FALSE (*fals diye okunur*) olarak yorumlanır. Sadece boolean – mantıksal değerler değil karakterler de sayı olarak işlem görebiliyor. ASCII karakterleri tamsayı olarak, mesela A'nın karşılığı olan 65 rakamını kullanabilirsiniz. 'A' + 2 * 'z' gibi... Bu arada 'A'<'a' işleminin sonucu 1 yani true çıkar. ASCII* kod tablosunda A'nın 65, a'nın 97 olduğu görülebilir. Türkçe karakterleri kullanmak için ASCII kod tablosunun ikinci sayfasından faydalanabilirsiniz.

```

if (ch>='a' && ch<='z' || ch=='ı' || ch=='ğ' || ch=='ü' || ch=='ç' || ch=='ö' || ch=='ş') {...}

```

İlginç kullanımlar:

```

i = ( j =2 , j + 3 );      // i = 5 , j = 2 işlem yapılırken atama yapılıyor
a = b = c = 50 ;          //hepsine 50 atanır
if(10<=x<=20) { }         //kabul edilir fakat en iyisi alttaki gibi yapmaktır:
if(10<=x && x<=20) { }     //&&, AND – ve anlamında
if( (ch=getchar() ) == 'q' ) {...} //tuş basımı ch'ye aktarılıp, q basılıp basılmadığına bakılır

```

Değişkenler int – tamsayı, float (*flout diye okunur*) – real (*riil diye okunur*) gerçel sayı , char (*çar diye okunur*) – karakter veya metin şeklinde kullanılabilir. Eğer 16 karakterli bir metin ayırmış isek, n-1 yani 15 karakterini biz kullanabiliyoruz. Son karakter "\0" yani NULL olarak saklanır. Demek ki metinler sonu "0" olan char dizileridir. Metinlere "=" ile atama yapılmaz; aşağıdaki örnek gibi **strcpy (degisken, deger);** yapılır. Karşılaştırmak için de strcmp (metin1, metin2) komutu kullanılır.

int x[6]; komutu hafızada altı adet iki byte tamsayı yer açılır. x[0], x[1], x[2], x[3], x[4] ve x[5] şeklinde kullanabiliriz. Hafızada diziler peş peşe oluşturulurken, peş peşe tanımlanan değişkenlerin artarda olacağı kesin değildir. Bu sebeple Basic dilindeki gibi Dim a (n) şeklinde boyutu belli olmayan dinamik dizi C dilinde tanımlanamaz. Dizilerin sonlandırıcı sabitleri C dilinde int a[6]; gibi her zaman belirtilmek zorundadır. Yani int a[]; olamaz. İndeks numaraları a[-5] gibi negatif de olamaz. Metinlerde ise char ad[] = "Ali"; olabilir.

Bu arada int değişkeni DOS ve Win3.1'de 16 bit yer kaplarken, 32 bit ortam olan Windows 9x'de 32 bit yer kaplar. Yani tamsayı değişkenlerinizi kullanırken işletim sistemi (Unix, Windows) ve işlemcinize göre boyu farklı olabilir. Aynı şekilde fakat daha az belirgin olarak da short ve long değişkenlerde de birkaç farklılaşma vardır. Bu türlerdeki uyumsuzluklara rağmen, char değişkeni (bir Byte) kullanmak en pratik ve ekonomik hale geliyor.

Kesirli sayıları saklayan float dört byte hafıza kullanırken, kesirli kısım farklı şekilde saklanır. IEEE – The Institute of Electrical and Electronics Engineers standardına göre kayar noktalı sayılar işaret kısmı, kesir kısmı ve üstel

* Ek 3 – ASCII Kod Tablosuna bakınız.

kısım olarak parçalar halinde saklanır. Bu sayıların bölme ve çarpma gibi karmaşık hesapları işlemcinin içindeki matematik işlemci kısmında yapılır.

Öğrenmek istediğiniz dildeki değişkenlerin hafızada kaç byte yer kapladığını (C dilinde sizeof() operatörü ile bunu bulabilirsiniz), değer aralıklarını ve çeşitlerini muhakkak inceleyin. Değişkenlerin sınır aralığını C dili kontrol etmediğinden, çalışmalarınızda garip rakamlarla karşılaşmamanız için boyutlarına dikkat edin. Yardım dosyalarında ayrıntılı bilgi bulabilirsiniz. Tamsayı, gerçel sayı ve metin tipleri arasındaki dönüşüm komutlarının kullanımı da önemlidir.

auto, register, static, extern, const ve volatile gibi yer ve tür bildirimleri vardır. Genellikle biz bunları kullanmadan değişkenlerimizi tanımlarız. İleri programcılar değişkenlerini daha düzenli hazırlarlar.

```
auto int a;           //zaten yerel değişkenler auto olarak varsayılır, stack segment'te tutulurlar
register int a;       //hafıza yerine sınırlı sayıda işlemci yazmacı kullanan yerel değişken
volatile int a;       //kesinlikle yazmaçlarda bu değişkeni bekletme, belleği kullan anlamındadır
static int a = 0;     //yerel ise her çağrılıştaki tekrar oluşturulmaz, değerini korur, data segment'te
extern int a;         //birleştirilen diğer programda da aynı değişkeni kullanabiliriz, fonksiyonları da extern ile dışardaki
                     //programdan alınacağı belirtilir
const int a = 10;     //değeri asla değiştirilemeyen sabit tanımlama

#include <stdio.h>
#include <string.h>
union tip {
    unsigned char ad [16];
    unsigned char soyad [16];
} kisiler;

main()
{
    strcpy( kisiler.soyad, "Dere" );
    strcpy( kisiler.ad, "Ahmet" );
    printf ( "%s\n" , kisiler.soyad );           //print ef diye okunur
}
```

Siz ekrana soyad yazacak diye beklerken ad yazıyor değil mi? Kazı ayağı demek öyle değilmiş. union - birlik sayesinde iki değişken aynı hafızayı kullanıyor. (Ornekler\C\union.c)

Type (C dilinde struct - structure) yani kullanıcı tanımlı tip, yapı oluşturma, Basic, C ve Pascal ortak özelliğe sahip; kisiler.soyad gibi tipin alt elemanı arasında nokta işareti kullanılır. İç içe yapı tanımlanabilir.

```
struct tarih{
    int gun;
    int ay;
    int yıl;
} eski_tarih, yeni_tarih;           //eski_tarih hafızada 6 byte yer kaplar
struct tarih ara_tarih = {12,4,2001}; //ilk değer tanımlanması
struct tarih *deneme;               //yapılar işaretçi de olabilir
...
gunler = yeni_tarih.gun - eski_tarih.gun;
(*deneme).ay = 12;                  //deneme->ay = 12; de denilebilir
```

Tüm değişkenler madem hafızaya yerleşiyor, onlara adres belirterek de ulaşabilirsiniz. Bunun anlamı pointer – işaretçi. Bir değişkenin adresi bir pointer değişken ile saklanarak doğrudan pointer üzerinden işlemlerimizi gerçekleştirebiliriz. Böylece yerel bir değişken tanımlasak da pointer sayesinde global olarak değerlere erişebiliriz. Yanımıza kalan hız ve bellek kazancı. C dilinde yoğun olarak kullanılan pointer yanlış kullanıldıklarında Windows'ta mavi ekran çıkmasına, geçersiz işlem hatalarına, DOS için ise reset – tekrar başlatma gibi etkilere sebep olur.

- ⌘ Bellek alanı ayrılmadan bir pointer'ın gösterdiği alana veri yüklemeyin.
- ⌘ Verileri saklarken ayrılan alanların büyüklüklerine dikkat edin.
- ⌘ Pointer değerlerinin geliş güzel değişimlerine izin vermeyin.
- ⌘ Dizilerin 1'den değil 0'dan başladığını unutmayın.

Bu arada pointer da bir değişken olduğuna göre ona da pointer (**), pointer pointer'a da pointer tanımlanabilir. Neyse boş verin! Bu kadarı da fazla :) İşaretçiler genellikle char (bir byte), int (iki byte) ve float (dört byte) tanımlanabilir.

```

int k;
int i; //tamsayı i bellekte ayrılır
int *iptr; //tamsayı iptr – integer pointer, int *iptr = &i; de olabilirdi
iptr = &i; //artık iptr i'nin adresine sahip
*iptr = 55; //aslında i'ye 55 değeri aktarıldı
k = *iptr //aslında i'nin değeri k'ya aktarıldı

...
char *hata = "Dosya yok!" //metin işaretçilere ilk değer atanabilir

...
iptr = (int *) 0x1b64; //Belli bir adres işaretçiye atanabilir, ama tehlikelidir (near pointer - yakın)
*iptr = 2001; //işaretçiye değer aktarımı
printf ("%p", iptr); //ekrana 1B64 yazar
printf ("%d", *iptr); //ekrana 2001 yazar
printf ("%x", &iptr); //ekrana fff2 yani işaretçinin adresini yazar

...
char far *cptr; //1 MB (0-FFFF) içinde tanımlanabilen işaretçi (far pointer - uzak)
p = (char far *) 0x124532bc //1245 segment, 32bc offset adresidir

...
char far *ekran = (char far *) 0xb8000000; //B800 DOS'un ekran bilgilerinin tutulduğu yerdir
*(ekran+20) = '.'; //metin ekranın ilk satırının 20. sütununa "." karakterini basar

...
int a[3] = {12,55,88}; //üç elemanlı dizi (0, 1 ve 2 numaralı)
int *iptr; //int *iptr = a; da olabilirdi
iptr = a; //dizi başlangıç adresi & ile gösterilemez
printf ("\n%d",*(iptr+1)); //ekrana 55 yazar, 0 1 2 indeks numaralıdır
printf ("\n%d", iptr[2]); //ekrana 88 yazar

...
void deneme (int *iptr) //fonksiyona bir değişkenin adresi gelecek
{ *iptr=100; } //işaretcinin gösterdiği yere değer aktarılır
void main(void)
{int x;
deneme (&x); //x değişkeninin adresi fonksiyona yollanır
printf ("%d", x);} //ekrana 100 yazar

...
char *ad_oku (void) //değer dönderen fonksiyon
{char s[50];
printf ("Adınız=");gets (s);
return s;} //okunan metin değer dönderilir
main()
{char *cptr;
cptr= ad_oku(); //dönecek değer ancak bir char işaretçiye aktarılabilir
printf ("%s", cptr);
}

...
main(){ // "Ornekler\C\hafiza.c"
unsigned char far *cptr = (char far *) 0;
int k=0;
for (;;)
{if (k==0)
{putchar ("x ile çıkılır...\n ");
if (getch()=='x') break;
printf ("%Fp", cptr);
k = 16;} //ekrana 16'şar bellek kümesi gelir
printf ("%02x", *cptr);
--k;
++p;
}}

```

Aşağıdaki ?: şart deyiminde dikkat, VB içinde de tek satırda iif (mantıksal ifade, Doğru, Yanlış) komutu

bulunuyor:

```
#include <stdio.h>
```

```
main()
```

```
{int i=4; //i hem tanımlanır hem de ilk değeri atanır
```

```
printf("%d",i=4?4:5); //eğer i değişkeni 4'e eşitse ekrana 4, değilse 5 yaz...
```

```
}
```

Özellikle menü hazırlarken, birden fazla tuş okunmaması için klavye hafızasının boşaltılması gerekebilir:

fflush (stdin); //klavye belleği sıfırlanır

printf fonksiyonunda kullanılan bazı sabitler:

\a	zil sesi, ASCII kodu 7	%d	tamsayı (char ve int)
\b	boşluk, ASCII kodu 8	%c	tek karakter (char)
\t	TAB – tabulator karakteri, ASCII kodu 9	%x	tamsayının heksadesimal hali (int)
\n	yeni satır, ASCII kodu 10	%f	kayar noktalı (float ve double)
\v	düşey TAB, ASCII kodu 11	%s	metin (char)
\r	satır başı, ASCII kodu 13		

scanf()'de birden fazla veri girişi yapılabilir:

```
int a,b;
scanf("%d%d",&a,&b);           //Enter, Tab veya boşluk ile iki tamsayı okunur
scanf("%d,%d",&a,&b);           //10,4 gibi sadece virgül ile iki tamsayı girişi

scanf("%s", ad);                //metinlerde &ad gibi & işareti kullanılmasına gerek yoktur, okunan metinde boşluk varsa sadece ilk
                                //kısm ad'a atanır, eğer gets (ad); kullanılırsa böyle bir sorun olmaz
```

Üç çeşit fonksiyon olabilir:

1. Değer döndürebilen:

```
int fonksiyon(void)              //void - boş okunaklılığı artırır
{
...
return deger;
}
```

2. Değer alan ve / veya değer döndürebilen: (pointer kullanımı ile birden fazla değer dönmesi sağlanabilir)

```
int fonksiyon(int a, int b)      //( int a,b ) olmaz
{
...
return deger;
}
```

3. Prosedür gibi, ne değer alan ne de değer döndürebilen:

```
void fonksiyon(void)
{
...
return;
}
```

Eğer fonksiyon main()'den sonra yazılmış ise, main() öncesinde rededlere – tekrar tanımlama yani prototip tanımlaması yapılmalıdır. Ama genellikle fonksiyonlarımızı main()'den önce yazmaya çalışmamız alışkanlık olmalıdır.

```
#include <stdio.h>
float bolme (double a, double b);           //fonksiyon prototipi
main()
{
printf ("%f", bolme (12.4, 4.3));
}
float bolme (double a, double b)
{
return a / b;
}
```

Dinamik hafıza kullanımı C için çok doğaldır: (Ornekler\C\malloc.c)

```
#include <stdio.h>
```



```
#include <stdlib.h>
#include <string.h>
void main (void)
{
    char *yazi;                //pointer yazi
    register int t;            //işlemci yazmacı kullan, daha hızlıdır, sadece int ve char için

    yazi= (char *) malloc (80); //memory allocate - hafıza ayır, text ekran 80x25
    if (!yazi)
        {printf ("Yetersiz hafıza...\n"); exit(1);}
    gets (yazi);                //boşluk dahil metin okuma
    for (t= strlen(yazi)-1; t>=0; t--)
        putchar (yazi[t]);
    free (yazi);                //hafızayı serbest bırak
}
```

Aşağıdaki örnek belli bir sayıyı bitisel işlemler kullanarak ikilik sisteme çeviriyor:

```
#include <stdio.h>
void main(void)
{
    int sayi = 25;
    int k, bit;
    for (k = 15 ; k>=0 ; --k)        //k- - yapmanın bir farkı yoktur
    {
        bit = (sayi >> k) & 1;      //Pascal karşıtı: bit:= (sayi shr k) and 1;
        printf("%d", bit);
    }
}
```

Aşağıda üstteki örneğin k üç iken, yani sonuçta ekrana gelecek olan 0001 1001 çevriminin dördüncü basamağının değerini bize veriyor. Son basamak sıfır sıra numarasına sahip olduğu için üçüncü değil dördüncü basamağa bakınız.

```
25 = (0001 1001)2                //mavi üç sayı sağa shift ediliyor, kaydırılıyor
25 >> 3 = (0000 0011)2
(25 >> 3) & 1 = (0000 0011)2 & (0000 0001)2 = (0000 0001)2 = 1        //ekrana 1 yazar
```

ESC tuşu basılana dek yazılanların kelime, harf, paragraf ve satır sayısını hesaplar ve rapor eder. if deyimine güzel bir örnek. (Ornekler\C\okul\vizesoru.c)

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main()
{int harf=-1; int satir=1; int paragraf=1; int kelime=1; int harfsay; char karakter;
  clrscr();
  do{karakter=getch(); printf("%c",karakter);
    if ((karakter!='.')&&(karakter!='\r')&&(karakter!=' ')) harf++;
    harfsay++;
    if ((karakter==' ')||(karakter==13)) kelime++;
    if (karakter==13) {paragraf++;satir++;printf("\n");};
  }while(karakter!=27);
  printf("\n%d=kelime sayısı ",kelime);
  printf("%d=harf sayısı ",harf);
  printf("%d=paragraf sayısı ",paragraf);
  printf("%d=satır sayısı \n ",satir);
  getch();
}
```

1. sıralama (balon, seçerek ve hızlı sıralama)

(Ornekler\C\sıralama\bubblsor.c, selesor.c, quicksor.c)

Dizilerde sıralama için en çok seçerek – selection sort, hızlı – quick sort ve balon – bubble sort kullanılıyor. Merge sort (birleştir sırala), insertion sort (ekleme sıralama) gibi algoritmalar da incelenebilir. Kullanım amacına göre hız gereken yerde hızlı sıralama, basit birkaç sayı sıralanacak ise de balon sıralaması kullanılabilir. Balon sıralamasında içteki for'dan geriye doğru giden bir tane daha yazılarak çift yönlü – bi-directional balon sıralaması olarak hızlandırılabilir. Pascal, Basic dillerinde de algoritma değişmiyor. Delphi'nin "Demo\Threads" dizininde güzel bir örnek var. Tabii ki sıralandıktan sonra dizinin ilk elemanı en küçük, son elemanı da en büyük rakam olur. Bu sayede minimum ve maksimum kendiliğinden ortaya çıkıyor. Büyük veri yığınlarında yani veritabanlarında ise SQL* deyimleri olan basit komutlar kullanarak istediğiniz alanı veya alanları sıralayabilirsiniz.

Balon sıralaması:

```
#include <stdio.h> // "bubblsor.c", basit ve yavaş algoritma
int a [10] = {33, 3, 6, 44, -3, 22, 12, 33, -4, 0};
void main (void)
{
    int k, l, tampon;
    for (k=0 ; k < 9 ; ++k)
        for (l = 0 ; l < 9 ; ++l )
            if (a[l] > a[l+1]) //tüm sayılar karşılaştırılarak yer
                //değiştirilir
                {
                    tampon = a[l];
                    a[l] = a[l+1];
                    a[l+1] = tampon;
                }
    for (k = 0 ; k < 10; ++k) printf (" %d", a[k]);
}
```

Seçerek sıralama:

```
#include <stdio.h> // "selesor.c", basit ve hızlı algoritma
int a [10] = {33, 3, 6, 44, -3, 22, 12, 33, -4, 0};
void main (void)
{
    int k, l, max, indis;
    for (k=0 ; k < 10 ; ++k)
    {
        max = a [k];
        indis = k;
        for (l = k + 1 ; l < 10; ++l )
            if (a[l] > max) //en büyük bulunarak başa atılır
            {
                max = a[l];
                indis = l;
            }
        a[indis] = a[k];
        a[k] = max;
    }
}
```

Ekrana çıkan sonuç:

-4 -3 0 3 6 12 22 33 33 44

Balon sıralamasının mikroişlemci kodları ise, şu şekilde not defterine yazılıp, debug'a yapıştırılabilir:

```
r ip
100
```

```
for (k = 9 ; k >= 0; --k) printf (" %d", a[k]);
}
```

Hızlı sıralama:

```
#include <stdio.h> // "quicksor.c", karmaşık ve hızlı algoritma
int a [10] = {33, 3, 6, 44, -3, 22, 12, 33, -4, 0};
void QuickSort(int a[ ], int lo0, int hi0)
{
    int lo = lo0; //lo0; le harfi, o harfi ve sıfır
    int hi = hi0;
    int mid, tampon;
    if ( hi0 > lo0 )
    {
        mid = a[ ( lo0 + hi0 ) / 2 ];
        while( lo <= hi )
        {
            while( ( lo < hi0 ) && ( a[lo] < mid ) ) ++lo;
            while( ( hi > lo0 ) && ( a[hi] > mid ) ) --hi;
            if( lo <= hi )
            {
                tampon = a [lo];
                a [lo] = a [hi];
                a [hi] = tampon;
                ++lo; --hi;
            }
        }
        if( lo0 < hi )
            QuickSort( a, lo0, hi );
        if( lo < hi0 )
            QuickSort( a, lo, hi0 );
    }
}

void main(void)
{ int k;
  QuickSort (a, 0, 9); //0 ilk sıra, 9 son sıra numarası
  for (k = 0 ; k < 10; ++k) printf (" %d", a[k]);
}
```

* Ek 2 – SQL kısmına bakınız.

```

r cs
1000
r ds
2000
e ds:100 33 3 6 44 3 22 12 33 4 0    negatif sayılar yazılamıyor
a 100
r ip
100
r cs
1000
r ds
2000
e ds:100 33 3 6 44 3 22 12 33 4 0
a 100
1000:0100    mov cx, 0009    Dizi boyutunun bir eksiği
1000:0103    mov si, 0100    Dizi başlangıç adresi
1000:0106    jmp 010d        Alt programa git
1000:0108    loop 0106       Ana döngü
1000:010A    int 20          Program sonu
1000:010C    nop            İşlem yok
1000:010D    push cx         cx değerini yeni döngüde koru
1000:010E    mov cx,0009     İç döngü de 10 – 1, yani 9
1000:0111    mov si,0100     Dizi başlangıç adresi
1000:0114    mov al, [si]     İlk değer
1000:0116    mov ah, [si+01]  Bir sonraki değer
1000:0119    cmp al, ah       İkisini karşılaştırır
1000:011B    ja 011F         Eğer al>ah (al büyükse) ise
1000:011D    jmp 0126        Değilse dönmeye devam et
1000:011F    xchg al,ah       İkisini yer değiştir
1000:0121    mov [si],al      Yeni değerleri güncelle
1000:0123    mov [si+01], ah  Bir sonraki değer
1000:0126    inc si          Bir sonraki adres
1000:0127    loop 0114       Alt döngü devam eder
1000:0129    pop cx          Eski cx ne idi?
1000:012a    jmp 0108        İkinci döngü bitiyor
g=cs:100 10a
d ds:100
2000:0100    00 03 03 04 06 12 22 33-33 44 00 00 00 00 00 00 ..... "33D.....
2000:0110    00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....

```

Mavi yazılı olan adresleri daha programı yazarken bilmemiz mümkün değil. Debug'a yapıştırdıktan sonra adreslere bakıp buraları düzeltiriz. Bu adres işlemi Microsoft Programmers Workbench adlı program ile veya başka dil içinde asm kodları kullanırken Label tanımlanarak yapılıyor. 0 – 255 arasında olan onaltılık sayı sistemindeki tamsayıları rahatlıkla sıralayabiliyoruz. Bu alan yetmezse 0 – 65565 arası değer alabilen ikişer byte'lık AX ve BX yazmaçlarını buradaki AL ve AH yerine kullanabiliriz. Fakat SI+1 yerine de SI+2 demek gerekecektir. C dilindeki *(iptr+1) şeklindeki pointer artırımında kendiliğinden sonraki dizi elemanına geçilir.

Dizi sıralamasından sonraki en çok karşılaşılan problem **arama**dır. Değer aramanın sık kullanılan üç çeşidi vardır: linear (doğrusal), binary (ikilik) ve hashing (kıyma).

1- Doğrusal arama çok basittir, ilk dizi elemanından başlayıp bulana dek karşılaştırmaya devam edilir. Çok büyük dizilerde bu yöntem yavaştır.

2- İkilik de ise **sıralı** olan bir dizide aranan değer ile dizinin ortasındaki değer karşılaştırılır. Eğer dizi 1000 elemanlı ise 500 değer artık elenmiş olur. Kalan yarıda da ortadaki eleman karşılaştırılarak 250, daha sonra da 125 eleman kalacak şekilde döngü devam eder. Burada recursive (kendini çağıran) veya iterative (tekrarlan) yöntemler kullanılabilir.

Recursive Binary Search Fonksiyonu: (Pascal versiyonu – BinSearch (x, 53, 1, 30); gibi kullanılıyor)

type

```
x=array[1..100]of integer; // kendi tipimizi oluşturuyoruz
```

```
function BinSearch (var Table: x; Target: integer; First,Last: integer ) : integer;
var
```

```

middle: integer;

begin
    middle:=(First+Last) div 2;           //tamsayı olmalı
    if First>Last then BinSearch:=0
    else if Target = Table[middle] then BinSearch:=middle
    else if Target < Table[middle] then BinSearch:= BinSearch (Table, Target, First, middle - 1)
    else if Target > Table[middle] then BinSearch:= BinSearch (Table, Target, middle + 1, Last);
end;

```

Iterative Binary Search Fonksiyonu:

```

type
    x=array[1..100]of integer; // kendi tipimizi oluşturuyoruz
function BinSearch (var Table: x; Target: integer; First,Last: integer ) : integer;
var
    middle: integer;
    found: Boolean;

begin
    found := False;
    While (First<=Last) and not(found) do //şart doğru olana dek aramaya devam et...
    Begin
        Middle:=(First+Last) div 2;
        if Target = Table[middle] then found:= true
        else if Target < Table[middle] then Last:= middle - 1
        else if Target > Table[middle] then First:=middle + 1 ;
    End;
    if found then BinSearch:=middle
    else BinSearch:=0;           //bulunamadı
End;

```

3- Hashing Search sıralı olmayan büyük dizilerde de bu yöntem tercih edilir. Microsoft Access’de bir tablo oluşturduğunuzda “primary key field” oluşturayım mı diye soruyor hatırlarsanız. Her kayda verilen bir anahtar alan sayesinde tek taramada istenen kayda ulaşılabilir ve tabloları ilişkilendirmek kolaylaşıyor. Silinen kayıtların sıra numarası daha sonra gerçek kayıtlar ile değiştiriliyor. Sıfır ile maksimum slot (kayıt) arasında her kayda bir hash index (tek numara) veriliyor. Aynı değerdeki kayıtlarda ise uygun algoritma oluşturularak collision (çakışma/çarpışma) meydana gelmesi önlenir.

Basitce Delphi ile doğrusal arama ve değiştirme örneği (metinBul.zip):

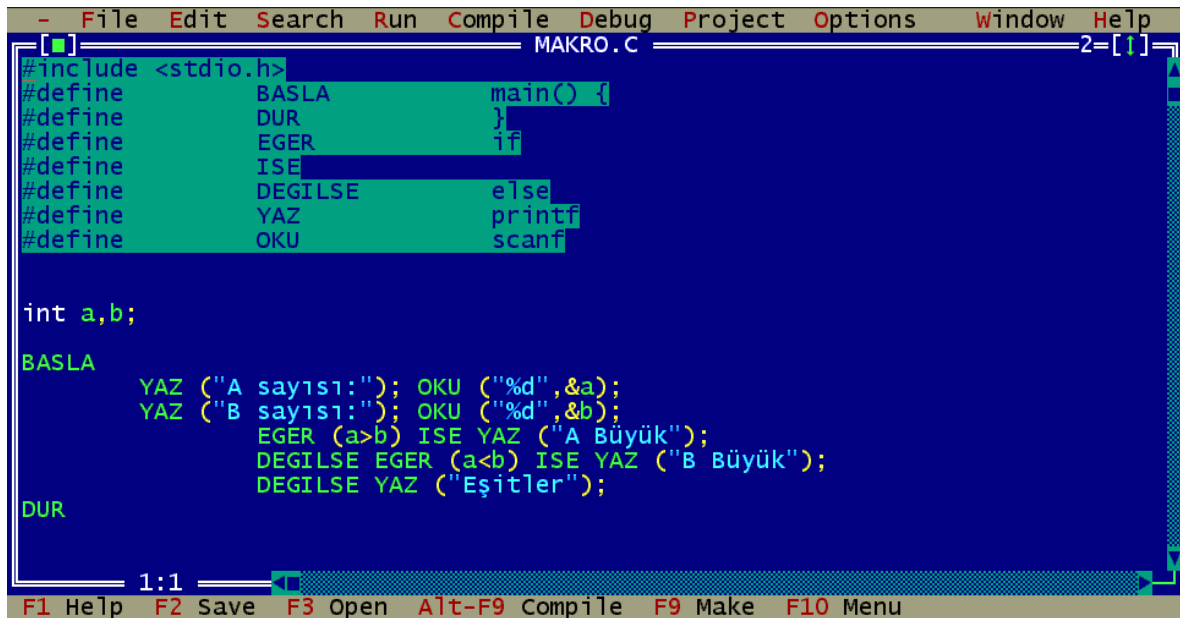
2. makro.c (#define kullanımı)

(Ornekler\C\ MAKRO.C)

```
#include <stdio.h>
int a,b;
main()
{
    printf("A sayısı:"); scanf("%d",&a);
    printf("B sayısı:"); scanf("%d",&b);
    if (a>b) printf("A Büyük");           //Pascal'dan farklı olarak ";" else öncesinde yazılabilir
        else if (a<b) printf("B Büyük");
        else printf("Eşitler");
}
```

Üstteki programı Türkçeleştirebiliriz. Prensip olarak makrolar büyük harf yazılır:

```
#include <stdio.h>
#define BASLA      main() {
#define DUR        }
#define EGER       if
#define ISE                //karşılığı yok...
#define DEĞİLSE      else
#define YAZ          printf
#define OKU           scanf
int a,b;
BASLA
    YAZ ("A sayısı:"); OKU ("%d",&a);
    YAZ ("B sayısı:"); OKU ("%d",&b);
    EGER (a>b) ISE YAZ ("A Büyük");
        DEĞİLSE EGER (a<b) ISE YAZ ("B Büyük");
        DEĞİLSE YAZ ("Eşitler");
DUR
```



Resim 2 – C dilinde basit makro yazımı

Ayrıca makrolar fonksiyon veya prosedür olarak da kullanılabilir:

```
#define dur()          while (! kbhit())          //dur();
#define diskriminant (a,b,c)  b * b - 4 * a * c    //sonuc=diskriminant(1,y,4);
```

3. unionreg.c (union tanımlama)

(Ornekler\C\unionreg.c)

Birkaç konuyu bir araya getirirsek, işlemcimizin yazmaçlarına ulaşmak için şunu kullanabilirsiniz:

```
Toplam disk alanı      = 1457664 B
Toplam dolu disk alanı = 1391616 B
Toplam bos disk alanı  = 66048 B
```

Resim 3 – Disket bilgilerinin görüntülenmesi

```
#include <stdio.h>
#include <dos.h>
#include <conio.h>
union REGS bizimreg;           //REGS yapısı dos.h içine tanımlı yapıdır.
double toplam, dolu, bos;
double topclus;
double bosclus;
double secsay;
main()
{
    clrscr();
    bizimreg.h.ah=0x36;
    bizimreg.h.dl=0;           //1,2,.. diger sürücülerdir.
    intdos(&bizimreg,&bizimreg);

    topclus= (unsigned) bizimreg.x.dx;
    bosclus= (unsigned) bizimreg.x.bx;
    secsay = (unsigned) bizimreg.x.ax;
    toplam = 512.0 * topclus * secsay;
    bos    = 512.0 * bosclus * secsay;
    dolu   = toplam - bos;

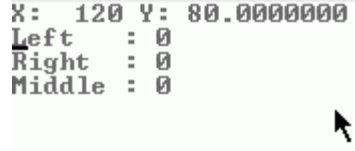
    printf("Toplam disk alanı    = %9.f B\n",toplam);
    printf("Toplam dolu disk alanı = %9.f B\n",dolu);
    printf("Toplam bos disk alanı  = %9.f B\n",bos);
    getch();
}
```

dl yada DL yazmacına 0 değeri yolladığımız zaman varsayılan sürücü olan "c:" raporlanır. 2 GB üstü için yanlış rapor verebilir. İsterseniz 1 değeri atayıp, disket bilgileri ile programı test edin. Byte yerine kilo byte olması için değerleri 1024'e bölebilirsiniz. O zaman printf'lerdeki "%9.f B" yerine "%9.2f KB" yazmanız gerekli.

4. mousetst.c (fare kullanımı)

(Ornekler\C\mousetst.c)

Bu program iki *.c dosyasından oluşuyor. mousetst.c ve mouse.c dosyası. Aşağıdaki kod mousetst.c 'dir.



```
X: 120 Y: 80.0000000
Left : 0
Right : 0
Middle : 0
```

Resim 4 – Fare testi

```
#include <conio.h>
#include <stdio.h>
#include "mouse.c"
void main ()
{
    //Check whether a mouse is connected
    if (!IfMouse) {
        printf ("Mouse not installed\n");
        return;
    }
    //Clear the screen
    clrscr ();
    //Activate the cursor
    ShowMouse ();
    do {
        //Get the mouse status
        ReadMouse ();
        //Display the coordinates
        gotoxy (1, 1);
        printf ("X: %4d Y: %4f\n", MouseX, (float)MouseY*2.5);
        //If MouseB & 1 == 1 then the left mouse button is down
        printf ("Left : %d\n", (MouseB & 1) == 1);
        //If MouseB & 2 == 2 then the right mouse button is down
        printf ("Right : %d\n", (MouseB & 2) == 2);
        //If MouseB & 4 == 4 then the middle mouse button is down
        printf ("Middle : %d\n", (MouseB & 4) == 4);
        //repeat that until both left and right buttons are down
    } while (!kbhit());
    //Disable the cursor
    HideMouse ();
}
```

mouse.c ise;

```
unsigned int MouseX, MouseY, MouseB;
char IfMouse ();
void ShowMouse ();
void HideMouse ();
void ReadMouse ();
void SetMouseXY (unsigned int X, unsigned int Y);
void SetMinMaxX (unsigned int Min, unsigned int Max);
void SetMinMaxY (unsigned int Min, unsigned int Max);

//This procedure checks where a mouse is available
char IfMouse ()
{
    unsigned int Result;
    asm mov ax, 0
    asm int 0x33
    asm mov Result, ax
    return Result;
}
```

```

}

//This procedure makes the mouse cursor visible
void ShowMouse ()
{
    asm mov ax, 0x1
    asm int 0x33
}

//This procedure makes the mouse cursor invisible
void HideMouse ()
{
    asm mov ax, 0x2
    asm int 0x33
}

//This procedure reads the location of the mouse cursor and the button press
//information
void ReadMouse ()
{
    asm mov ax, 0x3
    asm int 0x33
    asm mov MouseB, bx
    asm mov MouseX, cx
    asm mov MouseY, dx
}

//This procedure sets the mouse cursor to another location
void SetMouseXY (unsigned int X, unsigned int Y)
{
    asm mov ax, 0x4
    asm mov cx, X
    asm mov dx, Y
    asm int 0x33
}

//This procedure sets the minimum and maximum values in horizontal range
void SetMinMaxX (unsigned int Min, unsigned int Max)
{
    asm mov ax, 0x7
    asm mov cx, Min
    asm mov dx, Max
    asm int 0x33
}

//This procedure sets the minimum and maximum values in vertical range
void SetMinMaxY (unsigned int Min, unsigned int Max)
{
    asm mov ax, 0x8
    asm mov cx, Min
    asm mov dx, Max
    asm int 0x33
}

```

Her iki programı da aynı klasöre yazmanız gereklidir. Fakat mouse.c'yi daha başka programlarda da kullanmak isterseniz "TC\Include" dizinine bu dosyayı kopyalayabilir, ve başka programın içine

```
#include <mouse.c>

```

diye ekleyebilirsiniz.

Pascal'da da *.tpu unit oluşturarak "TP\TPU" dizinine genel program prosedür ve fonksiyonları oluşturabilirsiniz. diyet1.pas örneğine bakabilirsiniz. Aslında şunu yapmak mümkün; madem fareye mikroişlemci kodları ile ulaşıyor, bunu alıp Basic ve Pascal içinde de kullanabilirsiniz. Yani Pascal örneğimiz olan hesapmak.pas hesap makinesi fare ile de çalışsa ne güzel olurdu... "Ornekler\C\Grafikli.zip*grafikli.c" tam istediğimiz gibi, fare ve grafik ekran kullanımı ile ilgili...

5. klvy.c (klavye ile çizim yapma)

(Ornekler\C\klvy.c)

Assembler bölümündeki cizim.asm gibi bu da ekrana klavyeden çizim yapılıyor. w, a, s, d yön için, q çıkış, e silme, r çizim, f atlama, g bitişik pixel çizme için kullanılıyor. Dikkat ettiyseniz her iki *.c örneğimizde asm satırları var. Her dil donanıma doğrudan komut yollayabilmek için bu tip kodlanmaya gidiyor. Bu sayede grafik ekrana (320*200 çözünürlük) geçip renkli nokta basıyoruz. cizim.asm örneğinden farklı olarak ekranın dışına çıkılmaması için önlemler alınmış.



Resim 5 – Klavyeden çizim yapmak

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>
typedef unsigned char BYTE; //artık BYTE'ın anlamı unsigned char
typedef unsigned short WORD;
WORD SCREEN_WIDTH=320,SCREEN_ADDRESS=0xa000;
void putPixel(int x,int y,BYTE color)
{ asm mov ax,y
  asm mul SCREEN_WIDTH
  asm add ax,x
  asm mov bl,color
  asm xchg ax,bx
  asm mov es,SCREEN_ADDRESS
  asm mov es:[bx],al
}
main()
{ int x=10;int y=10;char ch;char erase,fast=1;
  clrscr();
  asm mov ax,13h
  asm int 10h

  putPixel(x,y,60);
  do
  {ch=getch();

  if (ch=='w') {if(erase)putPixel(x,y,0);else putPixel(x,y,98);
               if(fast)y=y-fast;else y--;
  if(x<=0) x=0;
  if(x>=300) x=300;
  if(y<=0) y=0;
  if(y>=200) y=199;
               putPixel(x,y,60);
               }
  if (ch=='s') {if(erase)putPixel(x,y,0);else putPixel(x,y,98);
               if(fast)y=y+fast;else y++;
  if(x<=0) x=0;
  if(x>=300) x=300;
  if(y<=0) y=0;
  if(y>=200) y=199;
               putPixel(x,y,60);
               }
  if (ch=='a') {if(erase)putPixel(x,y,0);else putPixel(x,y,98);
               if(fast)x=x-fast;else x--;
  if(x<=0) x=0;
```

```

if(x>=300) x=300;
if(y<=0) y=0;
if(y>=200) y=199;
    putPixel(x,y,60);
}
if (ch=='d') {if(erase)putPixel(x,y,0);else putPixel(x,y,98);
    if(fast)x=x+fast;else x++;
if(x<=0) x=0;
if(x>=300) x=300;
if(y<=0) y=0;
if(y>=200) y=199;
    putPixel(x,y,60);
}
if(ch=='q') {
    asm mov ax,3
    asm int 10h
    exit(1);}
if(ch=='e') erase=1;
if(ch=='r') erase=0;
if(ch=='f') fast++;
if(ch=='g') fast=0;
}while(1);                                //sonsuz döngü
}

```

Eğer ekrana bir nokta basabilmişseniz, artık çizgi, üçgen, daire, kare gibi şekiller, hatta kendi yazı tipinizi oluşturabilirsiniz. Ama her şeyi de çizerek yapmak yerine belli dokuları resim olarak alabilirsiniz. Mesela harfleri dışarıda hazırlayıp, istediğiniz yere yazılar yazabilir, logo yapabilirsiniz. MSPaint veya Paintbrush ile zaten bu çizim işleri yapıyoruz. Çıkan ürün bitmap – bit haritalama (sadece renk ve koordinat bilgileri saklanır) türündedir.

CorelDraw, AutoCAD, Flash ise vektörel çizim yapmamızı sağlayan programlardır. Vektörel çizimin avantajı zoom yani büyütülse bile bozulmamasıdır ve matematiksel olarak çizimler hesaplanıp kaydedildiği için dosya olarak da az yer kaplamalarıdır. Yazı yazmak için kullandığımız truetype font – gerçek yazı tipleri de aslında minik vektörel resimlerdir. WYSIWYG – *what you see is what you get* ne görürsen onu alırsın mantığı ile yazıcımızdan ekranda gördüğümüz yazıları küçültüp büyüterek rahatlıkla alabiliyoruz. Bol bol işlemler geri alınabilir, bu hafızada fazla yer işgal etmez. Flash ile ilgilenenlere Swish ve Flax programlarını da incelemelerini tavsiye ediyorum. Bitmap tabanlı program olan Adobe Photoshop programı kaliteli görüntü üretir ve filtrelemelerle resimler üzerinde gerçekçi efektlerle oynamamızı sağlar. Layer tekniği ile katman olarak oluşturulan resimlere müdahale etmek daha kolaylaşıyor.

Bilgisayardaki üç boyutlu dünya ise gittikçe daha gerçekçi olmaya devam ediyor. Eskiden render – tekrar oluşturma komutu ile uzun sürede elde edilebilen kaliteli iki boyutlu görüntüler, artık real – time gerçek zamanlı olarak oyunlarda ve sinemalarda görülebilir. Bump mapping – kabarık haritalama, texture – doku kaplama, ışıklandırma ve gölgelendirme, environmental bump mapping – çevresel haritalama, sis gibi özellikler artık standart. Fakat hala saç, toz, çamur, bitkiler, dinamik ışık etkisi, gerçek zamanlı modifikasyonlar (eğilme, kopma, yanma, erime), dinamik geometri hesabı bilgisayarlarımızı çok zorluyor.

İnternet’te biraz araştırırsanız oyun, tıbbi veya endüstriyel amaçla, siz de kendi bedava üç boyut editörlerinizi elde edebilirsiniz, Delphi gibi bir dilde örneğin GLScene nesneleri ile üç boyutlu tasarım programları yapabilirsiniz.

Biraz incelediğim şu programları da, üçüncü boyut ile ilgilenenlere Caligari TrueSpace, NewTek Lightwave, Metacreation’s Ray Dream Studio programlarını incelemelerini tavsiye ediyorum. Tümünde animasyon hazırlama ve materyal düzenleme imkanı var. CD’imizde bulabileceğiniz TrueSpace 3SE ile yerçekimi, rüzgar, elastiklik, katılık, kırılma gibi birçok efekt uygulanabiliyor. Ayrıca CD’de Blender, Amapi ve Kashmir gibi bedava birkaç yazılım daha bulabilirsiniz. Sonuçta *.avi veya *.mov film elde etmek yanında Export komutları ile projeleri diğer programların desteklediği türlere dönüştürebiliriz. Bu arada Bryce 3D programı da gerçekten çok özgün menülere sahip bir render ve animasyon programı. Macromedia Fireworks ile iki boyutta hoş yazı efektleri elde edebilirsiniz.

6. domgunu.c (tarihin gününü bulma)

(Ornekler\C\okul\domgunu.c)

Okulda yaptığımız haftanın hangi günü doğduğunuzu bulan case, for, printf, scanf kullanımına bir örnek.

```
#include <stdio.h>
char ay[12]={31,0,31,30,31,30,31,31,30,31,30,31};
int gun,ayi,yil,i;
double tgun;
unsigned long mgun;
main()
{printf("dogum yiliniz=");scanf("%d",&yil);
printf("dogum ayiniz =");scanf("%d",&ayi);
printf("dogum gununuz=");scanf("%d",&gun);
ay[1]=(yil%4)?28:29;
tgun=(yil-1)*365.0+(yil-1)/4;
for(i=0;i<(ayi-1);i++) tgun+=ay[i];
tgun+=gun; mgun=tgun/7;i=tgun-mgun*7;
printf("dogdugunuz gun\n");
switch(i){ case 1:printf("pazar");break;
           case 2:printf("pazartesi");break;
           case 3:printf("sali");break;
           case 4:printf("carsamba");break;
           case 5:printf("persembe");break;
           case 6:printf("cuma");break;
           case 0:printf("cumartesi");break;
        };
}
```

Sadece haftanın günü değil iki tarih arasındaki gün veya başka birimdeki fark da gerekli olabilir. "Ornekler\Pascal\Staj\Ariza.pas" programında dakika cinsinden tarih farkı hesaplamak zorundaydım. Bu programdan hesap kısmını çıkararak aşağıdaki örneği oluşturdum. Asıl programda iki tarih arasındaki farkı tam olarak dakika cinsinden alabiliyorken, örnekte gün içinde de geçen dakikalar hesaplanmıyor. Tarihler hatalı girilirse -111 gibi bir rakam üretiliyor. (Ornekler\Pascal\Diger\gun.pas)

```
type aylar=array [1..12] of integer;
const ay:aylar=(31,0,31,30,31,30,31,31,30,31,30,31);

function gun_sayisi(a1,b1,c1,a2,b2,c2:integer):longint; {tamsayi gun sayisi bulur}
var i,j:integer;artik_gun,aradeg,arasay:integer;
begin
gun_sayisi:=0;aradeg:=0;
if (c1>c2)or((b1>b2)and(c1=c2))or((a1>a2)and(b1=b2)and(c1=c2)) then
begin gun_sayisi:=-111;exit; end; {tarih kontrolu}
if (c1=c2)and(b1=b2) then {yil & ay esit ise}
begin aradeg:=a2-a1;gun_sayisi:=aradeg;exit; end;
if c1-c2=0 then begin {ayni yil ise}
if b1=2 then if (c1 mod 4)<>0 then ay[2]:=28 else ay[2]:=29;
for j:=b1+1 to b2-1 do begin
if j=2 then begin
artik_gun:=c1 mod 4;
if artik_gun<>0 then ay[j]:=28 else ay[j]:=29;
end; {if}
aradeg:=aradeg+ay[j];
end; {for}
aradeg:=aradeg+ay[b1]-a1+a2;
gun_sayisi:=aradeg;exit;
end;
if c2-c1=1 then begin {bitisik iki yil}
if b1<>12 then begin for j:=b1+1 to 12 do begin
if j=2 then begin
artik_gun:=c1 mod 4;
if artik_gun<>0 then ay[j]:=28 else ay[j]:=29;
```

```

        end; {if}
        aradeg:=aradeg+ay[j];
    end; {for}
    aradeg:=aradeg+ay[b1]-a1;
end
else aradeg:=aradeg+ay[b1]-a1;
if b2<>1 then begin for j:=1 to b2-1 do begin
    if j=2 then begin
        artik_gun:=c2 mod 4;
        if artik_gun<>0 then ay[j]:=28 else ay[j]:=29;
    end; {if}
    aradeg:=aradeg+ay[j];
end; {for}
    aradeg:=aradeg+a2;
end
else aradeg:=aradeg+a2;
gun_sayisi:=aradeg;exit;
end;
if c2-c1>1 then begin {farkli yillar}
    if b1<>12 then begin for j:=b1+1 to 12 do begin
        if j=2 then begin
            artik_gun:=c1 mod 4;
            if artik_gun<>0 then ay[j]:=28 else ay[j]:=29;
        end; {if}
        aradeg:=aradeg+ay[j];
    end; {for}
    aradeg:=aradeg+ay[b1]-a1;
end
else aradeg:=aradeg+ay[b1]-a1;
for i:=1 to c2-c1-1 do begin for j:=1 to 12 do begin
    if j=2 then begin
        artik_gun:=(c1+i) mod 4;
        if artik_gun<>0 then ay[j]:=28 else ay[j]:=29;
    end; {if}
    aradeg:=aradeg+ay[j];
end; {for j} end; {for i}
if b2<>1 then begin for j:=1 to b2-1 do begin
    if j=2 then begin
        artik_gun:=c2 mod 4;
        if artik_gun<>0 then ay[j]:=28 else ay[j]:=29;
    end; {if}
    aradeg:=aradeg+ay[j];
end; {for}
    aradeg:=aradeg+a2;
end
else aradeg:=aradeg+a2;
gun_sayisi:=aradeg;exit;
end;
end;

begin
writeln(gun_sayisi(30,7,2001,30,8,2001)*24*60); //24 saat 60 dakika ile çarpılıp kaç dakika olduğu bulunuyor
end.

```

Delphi ve VB içindeki hazır tarih fonksiyonları zaten bu tür karmaşık mantıktan bizi uzaklaştırıyor. Bir iki satır ile gün sayısı bulunabilir veya daha değişik istekler giderilebilir. PHP de ise:

```

//Kullanıcının son login tarihi ile bugün arasında kaç gün varmış?
$fark= floor( (mktime(0,0,0,date("m"),date("d"),date("y"))-
mktime(0,0,0, //saat kısmı
substr(mysql_result($result,$i,"sonGiris"),5,2), //ay
substr(mysql_result($result,$i,"sonGiris"),8,2), //gun
substr(mysql_result($result,$i,"sonGiris"),2,2)) //yıl
)/86400); //24*60*60
//İki tarihin aynı formatta olması için mktime() kullanılıyor

```

7. saat.c (DOS saati)

(Ornekler\C\TSR2 ve Ornekler\C\okul\SAAT.C)

mystr.c kodlarını bazı minik değişiklikler ile TC'de derleme imkanım oldu. DOS hafızasında kalıcı olarak çalışan bu saati istediğiniz an kapatacak tuş ise Shift+F10. TSR – Terminate and Stay Resident mantığı karışık gibi, ama belki siz bazı değişiklikler yapabilirsiniz. Hafif C++ kokuları yükseliyor... Windows XP'de ise artık TSR yok.

```
E:\KITAP\Ornekler\C\TSR2>mytsr
TSR yüklendi...
E:\KITAP\Ornekler\C\TSR2>
```

18:00:33

Resim 6 – TSR saat

```
#include <stdio.h>
#include <dos.h>
#include <string.h>
#ifdef __cplusplus
#define __CPPARGS ...
#else
#define __CPPARGS
#endif
void interrupt (*oldint8) (__CPPARGS);
void interrupt (*oldint9) (__CPPARGS);
int ch;
#define SH_F10 0x5D00 //kapatma tuşu Shift+F10
long far *tick = (long far *) 0x0000046c; //timer tick – saat tıklaması
char s[10] = {0,0,':',0,0,':',0,0,0,0};
long total_sec;
int sec, min, hour;

void _writes(int row, int col, char *str)
{
    char far *scrp = (char far *) 0xb8000000; //0xb800 text ekran hafızası
    scrp += row * 160 + col * 2;

    while (*str != NULL)
    {
        *scrp = *str++;
        scrp += 2;
    }
}

int get_last_buffer(void)
{
    char far *t1 = (char far *) 1050;
    if (*t1 != *(t1+2))
    {
        t1 += *t1 - 26;
        return *(int far *) t1;
    }
    return 0;
}

void f1(int a, char *str)
{
    str[0] = a / 10 + '0';
    str[1] = a % 10 + '0';
}

void interrupt newint8(__CPPARGS)
{
    oldint8();
    total_sec = *tick * 10 / 182;
    sec = total_sec % 60L;
    min = total_sec % 3600L / 60;
    hour = total_sec / 3600;
    f1(hour, s);
    f1(min, s+3);
    f1(sec, s+6);
}
```

```

        _writes(0, 72, s);
    }

void interrupt newint9(__CPPARGS)
{
    oldint9();
    ch = get_last_buffer();
    if (ch == SH_F10)
    {
        setvect(0x9, oldint9);
        setvect(0x8, oldint8);
        freemem(_psp);
    }
}

void main(void)
{
    printf("TSR yüklendi...\n");
    oldint8 = getvect(0x08);
    oldint9 = getvect(0x09);
    setvect(0x8, newint8);
    setvect(0x9, newint9);
    keep(0, 500);
}
//program bitti, ama etki TSR biçimde

```

Aşağıdaki örnek de yapı struct – structure (*strakçır diye okunur*) oluşturmak ile ilgili. Structure kullanımı Object Oriented Programming – nesneye dayalı programlama mantığına benziyor.



Resim 7 – Saat.c’nin sonucu

```

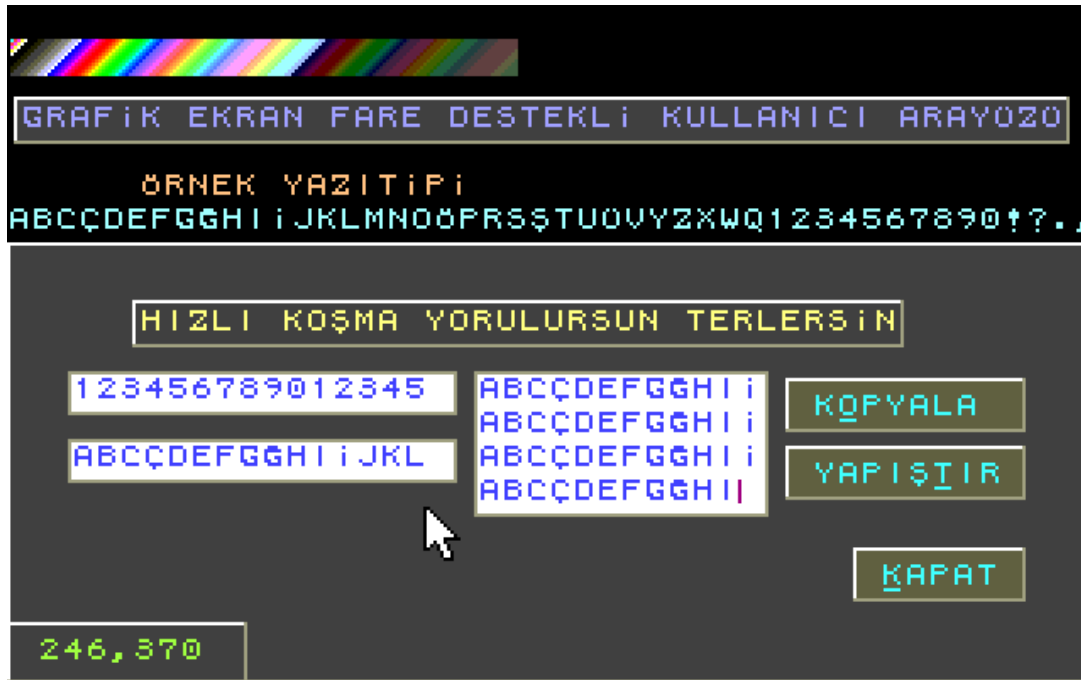
#include "stdio.h"
#include <conio.h>
#include <stdlib.h>
struct my_time{int hours,minutes,seconds;};
void display(struct my_time *t);
void update(struct my_time *t);
void delay(void);

void main(void)
{struct my_time systime;
 systime.hours=23;systime.minutes=59;systime.seconds=30;
 for(;;){update(&systime);display(&systime);}
}
void update(struct my_time *t)
{char cik;
 t->seconds++;
 if(t->seconds==60){t->seconds=0;t->minutes++;}
 if(t->minutes==60){t->minutes=0;t->hours++;}
 if(t->hours==24)t->hours=0;
 if(kbhit()) exit(0);
 delay();
}
void display(struct my_time *t)
{int i;
 clrscr();
 printf("%2d:",t->hours); printf("%2d:",t->minutes); printf("%2d\la",t->seconds);
}
void delay(void)
{long int t;
 for(t=1;t<15000000;+t);
}
//15.000.000 değeri ile oynanabilir

```

8. Grafikli.c (fareli grafik ekran)

(Ornekler\C\grafikli\grafikli.C)



Resim 8 – Ekran görüntüsü

Fare ve grafik ekran komutlarını birleştirerek, basit bir uygulama için gerekli form yapısını oluşturuyoruz. Ayrıca kendi yazı tipimizi, label (etiket), button (düğme), edit (metin kutusu), memo (çok satırlı veri girişi) nesnelerimizi de modüler bir yapıda hazırlıyoruz. Ana program olan grafikli.c dosyasına faregraf.c, nesneler.c ve yazitipi.c adlı üç program dahil edilerek yapıyoruz. Metin kutusu yaparken bayağı zorlandım, özellikle basit string – metin girişleri için garip kodlar yazmak biraz zor geliyor. Metin kutuları daha esnek olabilir. Mesela oklarla içinde dolaşarak Delete, Insert, Home ve Backspace tuşları daha etkili kullanılabilir. Düğmelere de basılma efekti verilebilir. Daire gibi özellikler için trigonometri komutları gerekebilir. Eski bir müzik çalma programında ekranın scroll – kaydırılması gibi ilginç bir kullanım görmüştüm. Siz de fare ekran kenarına geldiğinde kaydırma ile dar ekrandan sınırsız ekrana ulaşabilirsiniz. “Ornekler\Diger\FIash\hunt\hunting fla” örneğinde ekran kaydırma ile panoramik görüntü efekti elde edebiliyoruz.

Ek olarak listbox (liste kutusu), combobox (açılır kutu), checkbox (onay kutusu), radiobutton (radyo düğmesi) ve benzeri yapılabilir. Sağ tuş menüsü (altta kalan bölge getPixel (x,y) deyimi ile ön belleklenebilir) ve ana menü gibi daha profesyonel nesneler de çok hoş olabilir. İsterseniz dışarıdan *.bmp, *.gif gibi dosyaları da programa dahil edebilirsiniz. Yazı tipleri için ölçeklendirme ile daha küçük veya daha büyük font büyüklükleri de elde edilebilir. Buradaki örnekte yazı tipleri bitisel işlemlerle oluşturulurken, çizgiler ve eğrilerle yaptığınız kendi yazı tipinizi ölçeklendirmeyi deneyebilirsiniz. Bir harf beş sütun, sekiz satır bilgisinden oluşuyor. Her satırın ikilik düzendeki değeri bulunarak pixel konulup konulmayacağına karar veriliyor. En alt satır alt çizgi için kullanılıyor. Aslında alt çizginin fontu bir parçası olması, yeni font oluştururken iki kat daha fazla kod anlamına geliyor. İsterseniz alt çizgiyi sadece basit ayrı bir çizgi olarak da yapabilirsiniz. Bu arada özel karakterler ve küçük harfleri eklemedim, bunları da siz yapabilirsiniz. Windows yazı tiplerini resim olarak kaydedip istediğiniz yere yazabilmenizi sağlayan programları internetten bulabilirsiniz. Sadece bu yazı tipi konusunu bile ilerletip, yazı karakteri tanıma, iris tanıma, parmak izi tanıma gibi ticari konularda da kendinizi geliştirebilirsiniz.

Drag – drop (sürükle bırak) özelliği için de ekrandan resim bilgisi olarak pixel okuyup tampon bellekte saklamayı düşünebilirsiniz. Farenin etrafında oluşan istenmeyen görüntüler için de internetten fare temizleme komutları bulabilirsiniz. Daha yüksek ekran çözünürlükleri ve yazıcı kullanımını öğrenebilirsiniz. İmlecin yanıp sönmesini yapamadım ama bunu da siz yapabilirsiniz.

Sadece fare değil klavye kısayolları da etkin. Metin kutuları arasında TAB – tabulator (sekme) ile ve fare ile tıklayıp dolaşılabilir. Düğmelerin TAB duraklarını oluşturmam. ESC tuşu ve Kapat düğmesi (Alt+K) programı kapatır. TurboHelp programı içinde ileri seviye klavye kısayollarının kodlarını öğrenebilirsiniz. Alt, Ctrl ve Shift tuşları

tek başlarına kod üretemezler. Önce sıfır değeri, sonra da basılan diğer tuşa göre ikinci bir kod üretiliyor. Fare ile seçili olan metin kutusunun iç yazısı, Kopyala ve Yapıştır düğmeleri ile etiket deneme yazısının olduğu yere yapıştırılabilir. Alt köşede farenin gerçek x ve y koordinatları sürekli güncelleniyor. Ses ile geri besleme sağladım, ama yine de aşırıya kaçmamak gerekli. Her an bip sesleri zamanla sıkmaya başlıyor. Ama piyano yapabilirsiniz...

faregraf.c dosyası:

```
unsigned int MouseX, MouseY, MouseB;
// fare ve grafik komutlarının dosyası
void ShowMouse ()
{ asm mov ax, 0x1
  asm int 0x33
}
void HideMouse ()
{ asm mov ax, 0x2
  asm int 0x33
}
void ReadMouse ()
{ asm mov ax, 0x3
  asm int 0x33
  asm mov MouseB, bx
  asm mov MouseX, cx
  asm mov MouseY, dx
}
void SetMouseXY (unsigned int X, unsigned int Y)
{ asm mov ax, 0x4
  asm mov cx, X
  asm mov dx, Y
  asm int 0x33
}
void SetMinMaxX (unsigned int Min, unsigned int Max)
{ asm mov ax, 0x7
  asm mov cx, Min
  asm mov dx, Max
  asm int 0x33
}
void SetMinMaxY (unsigned int Min, unsigned int Max)
{ asm mov ax, 0x8
  asm mov cx, Min
  asm mov dx, Max
  asm int 0x33
}
void GrafikEkran(void) //320*200 ekran
{ asm mov ax,13h
  asm int 10h
}
void TekstEkran(void) //80*25 ekran
{asm mov ax,3
  asm int 10h
}
void EkranSil(void)
{asm mov cx,32000
  asm mov es,ADRES
  asm sub di,di
  asm sub ax,ax
  asm cld
  asm rep stosw
}
char getPixel(int x, int y)
{asm mov ax,y
  asm mul GENISLIK
  asm add ax,x
  asm mov bx,ax
  asm mov es,ADRES
```

```
asm mov al,es:[bx]
return _AL; //return boşluk alt çizgi büyük harf AL
}
void videoBekler(void)
{ asm mov dx,0x3da
  loop1:
  asm in al,dx
  asm test al,8
  asm jnz loop1
  loop2:
  asm in al,dx
  asm test al,8
  asm jz loop2
}
```

```
void ses(int hertz, int sure)
{
  sound (hertz);
  delay (sure);
  nosound ();
}
```

nesneler.c dosyası:

```
struct etiket{
  char *yazi; //iç yazı
  int basx; //başlangıç x
  int basy; //başlangıç y
  char renk; //yazı rengi
  short cerv;
  //çerçeve var ise 1(arkaplan=mor), yok ise
  0(arkaplan=transparan)
};
struct dugme{
  char *yazi; //iç yazı
  int basx; //başlangıç x
  int basy; //başlangıç y
  int sonx; //son x
  int sony; //son y
};
struct metin{
  char *yazi; //iç yazı
  int basx; //başlangıç x
  int basy; //başlangıç y
  char renk; //yazı rengi
  short harfsayisi; //kaç harf var
};
struct memo{
  char *yazi; //iç yazı
  int basx; //başlangıç x
  int basy; //başlangıç y
  char renk; //yazı rengi
  short harfsayisi; //kaç harf var
  short satsay; //kaç satırlık
};
void kutu2(int x1,int y1,int x2,int y2) //içi beyaz renkli
{ int i,j;
  for (i=x1;i<=x2;i++) putPixel(i,y1,28); // -
  for (i=y1;i<=y2;i++) putPixel(x1,i,28); // |
  for (i=x1;i<=x2;i++) putPixel(i,y2,28); // _
  for (i=y1;i<=y2;i++) putPixel(x2,i,28); // |
```



```

for (i=x1+1;i<x2;i++)
for (j=y1+1;j<y2;j++) putPixel(i,j,15);
}
void kutu1(int x1,int y1,int x2,int y2) //içi mor renk
{ int i,j;
for (i=x1;i<=x2;i++) putPixel(i,y1,15); // -
for (i=y1;i<=y2;i++) putPixel(x1,i,15); // |
for (i=x1;i<=x2;i++) putPixel(i,y2,28); // _
for (i=y1;i<=y2;i++) putPixel(x2,i,28); // |
for (i=x1+1;i<x2;i++)
for (j=y1+1;j<y2;j++) putPixel(i,j,154);
}
void kutu(int x1,int y1,int x2,int y2) //içi gri renkli
{ int i,j;
for (i=x1;i<=x2;i++) putPixel(i,y1,15); // -
for (i=y1;i<=y2;i++) putPixel(x1,i,15); // |
for (i=x1;i<=x2;i++) putPixel(i,y2,28); // _
for (i=y1;i<=y2;i++) putPixel(x2,i,28); // |
for (i=x1+1;i<x2;i++)
for (j=y1+1;j<y2;j++)
putPixel(i,j,24);
}
void karakter_bas (char yazi[ ],int basx,int basy,int harfno,
int aralik,char renk)
{switch (yazi[harfno])
{
case 'A':harfBas(_A,basx+aralik,basy,renk);break;
case 'B':harfBas(_B,basx+aralik,basy,renk);break;
case 'C':harfBas(_C,basx+aralik,basy,renk);break;
case 'Ç':harfBas(_Ch,basx+aralik,basy,renk);break;
case 'D':harfBas(_D,basx+aralik,basy,renk);break;
case 'E':harfBas(_E,basx+aralik,basy,renk);break;
case 'F':harfBas(_F,basx+aralik,basy,renk);break;
case 'G':harfBas(_G,basx+aralik,basy,renk);break;
case 'Ğ':harfBas(_YG,basx+aralik,basy,renk);break;
case 'H':harfBas(_H,basx+aralik,basy,renk);break;
case 'I':harfBas(_I,basx+aralik,basy,renk);break;
case 'İ':harfBas(_Ii,basx+aralik,basy,renk);break;
case 'J':harfBas(_J,basx+aralik,basy,renk);break;
case 'K':harfBas(_K,basx+aralik,basy,renk);break;
case 'L':harfBas(_L,basx+aralik,basy,renk);break;
case 'M':harfBas(_M,basx+aralik,basy,renk);break;
case 'N':harfBas(_N,basx+aralik,basy,renk);break;
case 'O':harfBas(_O,basx+aralik,basy,renk);break;
case 'Ö':harfBas(_Oo,basx+aralik,basy,renk);break;
case 'P':harfBas(_P,basx+aralik,basy,renk);break;
case 'R':harfBas(_R,basx+aralik,basy,renk);break;
case 'S':harfBas(_S,basx+aralik,basy,renk);break;
case 'Ş':harfBas(_Sh,basx+aralik,basy,renk);break;
case 'T':harfBas(_T,basx+aralik,basy,renk);break;
case 'U':harfBas(_U,basx+aralik,basy,renk);break;
case 'Ü':harfBas(_Uu,basx+aralik,basy,renk);break;
case 'V':harfBas(_V,basx+aralik,basy,renk);break;
case 'Y':harfBas(_Y,basx+aralik,basy,renk);break;
case 'Z':harfBas(_Z,basx+aralik,basy,renk);break;
case 'Q':harfBas(_Q,basx+aralik,basy,renk);break;
case 'W':harfBas(_W,basx+aralik,basy,renk);break;
case 'X':harfBas(_X,basx+aralik,basy,renk);break;
case '1':harfBas(_1,basx+aralik,basy,renk);break;
case '2':harfBas(_2,basx+aralik,basy,renk);break;
case '3':harfBas(_3,basx+aralik,basy,renk);break;
case '4':harfBas(_4,basx+aralik,basy,renk);break;
case '5':harfBas(_5,basx+aralik,basy,renk);break;
case '6':harfBas(_6,basx+aralik,basy,renk);break;
case '7':harfBas(_7,basx+aralik,basy,renk);break;
case '8':harfBas(_8,basx+aralik,basy,renk);break;

```

```

case '9':harfBas(_9,basx+aralik,basy,renk);break;
case '0':harfBas(_0,basx+aralik,basy,renk);break;
case '.':harfBas(_dot,basx+aralik,basy,renk);break;
case ',':harfBas(_comma,basx+aralik,basy,renk);break;
case '!':harfBas(_excl,basx+aralik,basy,renk);break;
case '?':harfBas(_ques,basx+aralik,basy,renk);break;
};

```

```

void etiket_nesnesi1 (struct etiket et)
{int i,aralik;
aralik=0;
if(et.cerv==1) kutu1(et.basx-3,et.basy-3,et.basx+(strlen(et.yazi)*7),et.basy+10);
for(i=0;i<=(strlen(et.yazi));i++)
{
karakter_bas (et.yazi,et.basx, et.basy,i,aralik,et.renk);
aralik+=7;
}
}

```

```

void etiket_nesnesi (char yazi[],int basx,int basy,char renk,short cerv)
{int i,aralik;
aralik=0;
if(cerv==1) kutu1(basx-3,basy-3,basx+(strlen(yazi)*7),basy+10);
for(i=0;i<=(strlen(yazi));i++)
{
karakter_bas (yazi,basx, basy,i,aralik,renk);
aralik+=7;
}
}

```

```

void dugme_nesnesi1(struct dugme dug)
{
kutu(dug.basx,dug.basy,dug.sonx,dug.sony);
etiket_nesnesi(dug.yazi,dug.basx+2+7,dug.basy+5,11,0);
}

```

```

void metin_nesnesi1(struct metin met)
{
kutu2(met.basx-2,met.basy-2,met.basx+met.harfsayisi*7,met.basy+10);
etiket_nesnesi(met.yazi,met.basx,met.basy,met.renk,0);
}

```

```

void memo_nesnesi1(struct memo me)
{int k,l,m,satno;
satno=1;m=1;
kutu2(me.basx-2,me.basy-2,me.basx+me.harfsayisi*7,me.basy+me.satsay*10);
if (strlen(me.yazi)<=12)
etiket_nesnesi(me.yazi,me.basx,me.basy,me.renk,0); else
{
for (l=0;l<strlen(me.yazi);)
{
for (k=l;k<(l+12);k++)
{
m++;
karakter_bas(me.yazi,me.basx+m*7-14,me.basy+satno*10-10,k,0,me.renk);
}
satno++;l+=12;m=1;
}
}
}

```



```

dugme_1.sonx=280+20;      dugme_1.sony=175;
dugme_nesnesi1(dugme_1);
harfBas(_UO,230+2+7*2,110+5,11); //altı çizili O
harfBas(_UT,230+2+7*6,130+5,11); //altı çizili T
harfBas(_UK,250+2+7,160+5,11); //altı çizili K
etiket_1.yazi="GRAFİK EKRAN FARE DESTEKLİ
KULLANICI ARAYŞIŞ";
etiket_1.basx=5;          etiket_1.basy=30;
etiket_1.renk=80;         etiket_1.cerv=1;
etiket_nesnesi1(etiket_1);
etiket_1.yazi="™RNEK YAZIT™P™";
etiket_1.basx=40;         etiket_1.basy=50;
etiket_1.renk=90;         etiket_1.cerv=0;
etiket_nesnesi1(etiket_1);

etiket_1.yazi="ABCDEFHGHIJKLMNO™PRS□ TUŞVYZXWQ
1234567890!?,.";
etiket_1.basx=1;          etiket_1.basy=60;
etiket_1.renk=100;        etiket_1.cerv=0;
etiket_nesnesi1(etiket_1);
etiket_1.yazi="HIZLI KO□ MA YORULURSUN TERLERS™N";
etiket_1.basx=40;         etiket_1.basy=90;
etiket_1.renk=93;         etiket_1.cerv=1;
etiket_nesnesi1(etiket_1);
metin_1.yazi=metin1;
metin_1.basx=20;          metin_1.basy=110;
metin_1.renk=9;           metin_1.harfsayisi=16;
metin_nesnesi1(metin_1);
metin_1.yazi=metin2;
metin_1.basx=20;          metin_1.basy=130;
metin_1.renk=9;           metin_1.harfsayisi=16;
metin_nesnesi1(metin_1);
memo_1.yazi=metin3;
memo_1.basx=140;          memo_1.basy=110;
memo_1.renk=9;            memo_1.harfsayisi=12;
memo_1.satsay=4;
memo_nesnesi1(memo_1);
imlec(1,girilen_harf1,0,5);
imlec(2,girilen_harf2,0,15);
imlec(3,girilen_harf3x,girilen_harf3y,15);
}

void yapistir(void)
{
etiket_1.yazi="";
etiket_1.basx=40;          etiket_1.basy=90;
etiket_1.renk=93;          etiket_1.cerv=1;
etiket_nesnesi1(etiket_1);
etiket_1.yazi=bellek;
etiket_1.basx=40;          etiket_1.basy=90;
etiket_1.renk=93;          etiket_1.cerv=0;
etiket_nesnesi1(etiket_1);
}

void kopyala (void)
{if (aktif_nesne_no==1) bellek=metin1; else bellek=metin2;
}

main()
{aktif_nesne_no=1;
metin1="123456789012345";
metin2="ABCDEFHGHIJKL";

metin3="ABCÇDEFGĞHİİABCÇDEFGĞHİİABCÇDEFGĞHİİA
BCÇDEFGĞHİİ";
girilen_harf1=15;

```

```

girilen_harf2=15;
girilen_harf3=48;
girilen_harf3x=12;
girilen_harf3y=4;
GrafikEkran(0x13); //320*200*256 ekran
refresh();
ShowMouse(); SetMouseXY(300,100);
do {
videoBekler(); //kırpılmayı önler
itoa(MouseX, ara1, 10);
itoa(MouseY*2.5, ara2, 10);
strcat(ara1,"");strcat(ara1,ara2);
kutu1(0,182,70,199); //status bar gibin
etiket_1.yazi=ara1;
etiket_1.basx=10;          etiket_1.basy=187;
etiket_1.renk=70;          etiket_1.cerv=0;
etiket_nesnesi1(etiket_1);
ReadMouse ();
if ((MouseB & 1) == 1 && MouseX>=38 && MouseX<=262
&&
((float)MouseY*2.5)>=272 && ((float)MouseY*2.5)<=300)
{ ses(100,1);aktif_nesne_no=1; }
if ((MouseB & 1) == 1 && MouseX>=38 && MouseX<=262
&&
((float)MouseY*2.5)>=322 && ((float)MouseY*2.5)<=350)
{ ses(100,1);aktif_nesne_no=2; }
if ((MouseB & 1) == 1 && MouseX>=278 && MouseX<=446
&&
((float)MouseY*2.5)>=272 && ((float)MouseY*2.5)<=445)
{ ses(100,1);aktif_nesne_no=3; }

if ((MouseB & 1) == 1 && MouseX>=462 && MouseX<=598
&&
((float)MouseY*2.5)>=277 && ((float)MouseY*2.5)<=310)
{ ses(900,90); kopyala();}
if ((MouseB & 1) == 1 && MouseX>=462 && MouseX<=598
&&
((float)MouseY*2.5)>=327 && ((float)MouseY*2.5)<=360)
{ ses(900,90); yapistir();}
if ((MouseB & 1) == 1 && MouseX>=502 && MouseX<=598
&&
((float)MouseY*2.5)>=402 && ((float)MouseY*2.5)<=435)
{ ses(500,10); goto cikis; }

if (kbhit()){
tus=getch();
if (tus==0) //Alt tuşu basılı olanlar için
{tus=getch();
switch (tus)
{case 20: //ve T basılıysa (Yapıştır)
ses(900,90);yapistir();break;
case 24: //ve O basılıysa (Kopyala)
ses(900,90);kopyala();break;
case 37: //ve K basılıysa (Kapat)
ses(500,10);goto cikis;break;
}
}
else
{
ses(50,1); //tuş feedback
if (tus==27) goto cikis; //ESCAPE
if (tus==9) //TAB
{
if (aktif_nesne_no==1) aktif_nesne_no=2; else
if (aktif_nesne_no==2) aktif_nesne_no=3; else
if (aktif_nesne_no==3) aktif_nesne_no=1;
}
}
}
}

```

```

} else
if(tus==8)          //BACKSPACE
{
switch (aktif_nesne_no)
{case 1:
if(girilen_harf1>0)
{ metin1[--girilen_harf1]=' ';
metin_1.yazi=metin1;
metin_1.basx=20;
metin_1.basy=110;
metin_1.renk=9;          metin_1.harfsayisi=16;
metin_nesnesi1(metin_1);
};
break;
case 2:
if(girilen_harf2>0)
{ metin2[--girilen_harf2]=' ';
metin_1.yazi=metin2;
metin_1.basx=20;          metin_1.basy=130;
metin_1.renk=9;          metin_1.harfsayisi=16;
metin_nesnesi1(metin_1);
};
break;
case 3:
if(girilen_harf3>0)
{ metin3[--girilen_harf3]=' ';
memo_1.yazi=metin3;
memo_1.basx=140;          memo_1.basy=110;
memo_1.renk=9;          memo_1.harfsayisi=12;
memo_1.satsay=4;
memo_nesnesi1(memo_1);
};
break;
}
}
else
{
switch (aktif_nesne_no)
{case 1:
if (girilen_harf1<15)
{ metin1[girilen_harf1++]=tus;
metin_1.yazi=metin1;
metin_1.basx=20;          metin_1.basy=110;
metin_1.renk=9;          metin_1.harfsayisi=16;
metin_nesnesi1(metin_1);
}
break;
case 2:
if (girilen_harf2<15)
{ metin2[girilen_harf2++]=tus;
metin_1.yazi=metin2;
metin_1.basx=20;          metin_1.basy=130;
metin_1.renk=9;          metin_1.harfsayisi=16;
metin_nesnesi1(metin_1);
}
break;
case 3:
if (girilen_harf3<12*4)
{ metin3[girilen_harf3++]=tus;
memo_1.yazi=metin3;
memo_1.basx=140;          memo_1.basy=110;

```

```

memo_1.renk=9;          memo_1.harfsayisi=12;
memo_1.satsay=4;
memo_nesnesi1(memo_1);
}
break;
}
} //girilen_harf
} //getch else'si
} //kbhit
if (aktif_nesne_no==1)
{imlec(1,girilen_harf1,0,5);
imlec(2,girilen_harf2,0,15);
girilen_harf3x=girilen_harf3 % 12;
if (girilen_harf3>=1 && girilen_harf3<=12)
girilen_harf3y=1;
if (girilen_harf3>=13 && girilen_harf3<=24)
girilen_harf3y=2;
if (girilen_harf3>=25 && girilen_harf3<=32)
girilen_harf3y=3;
if (girilen_harf3>=33 && girilen_harf3<=48)
girilen_harf3y=4;
imlec(3,girilen_harf3x,girilen_harf3y,15);
}
if (aktif_nesne_no==2)
{imlec(1,girilen_harf1,0,15);
imlec(2,girilen_harf2,0,15);
girilen_harf3x=girilen_harf3 % 12;
if (girilen_harf3>=1 && girilen_harf3<=12)
girilen_harf3y=1;
if (girilen_harf3>=13 && girilen_harf3<=24)
girilen_harf3y=2;
if (girilen_harf3>=25 && girilen_harf3<=32)
girilen_harf3y=3;
if (girilen_harf3>=33 && girilen_harf3<=48)
girilen_harf3y=4;
imlec(3,girilen_harf3x,girilen_harf3y,15);
}
if (aktif_nesne_no==3)
{imlec(1,girilen_harf1,0,15);
imlec(2,girilen_harf2,0,15);
girilen_harf3x=girilen_harf3 % 12;
if (girilen_harf3>=1 && girilen_harf3<=12)
girilen_harf3y=1;
if (girilen_harf3>=13 && girilen_harf3<=24)
girilen_harf3y=2;
if (girilen_harf3>=25 && girilen_harf3<=32)
girilen_harf3y=3;
if (girilen_harf3>=33 && girilen_harf3<=48)
girilen_harf3y=4;
if (girilen_harf3x>=1 && (girilen_harf3 % 12)==0 &&
girilen_harf3!=48)
girilen_harf3y++;
if (girilen_harf3==48) girilen_harf3x=149;
if (girilen_harf3==1) girilen_harf3x=1;
imlec(3,girilen_harf3x,girilen_harf3y,5);
}
} while (1);
cikis:
HideMouse ();
TekstEkran();
}

```

Bölüm 4 – Pascal

“Ornekler\Pascal\Staj\Hakanin” dizinindeki Takip.pas, “Ornekler\Pascal\Staj” dizinindeki Ariza.pas ve “Ornekler\Pascal\Prince3D” dizinindeki prince3d.pas dosyaları üzerinde iyi çalışılmış projeler. İncelemenizi tavsiye ederim.

**ARAŞTIRMACI GİRİSLERİ
PROGRAMI**

ARAŞTIRMACININ

**TAMAM
VAZGEÇ**

ADI :

SOYADI :

ESERİ :

KONUSU :

BASLIK :

GELDİĞİ TARİH :

İŞLEME GİRDİĞİ TARİH :

NOTLAR 1 :

NOTLAR 2 :

ÇIKMAK İÇİN (ESC) TUSUNA BASINIZ

ESC-Ana Menu/eski deger

Ariza Girisi

Telefon Numarasi = 1234567.

Sikayet Nedeni = soylemedi!

Sikayet Tarihi = 30/07/2001

Sikayet Saati = 17:12

Bilgi Telefonu = 1234568.

TAMAM

DUZELTME
KAYDETMEYEN ÇIK
KAYDET

Kayıt Sayısı=0



Resim 1 – Çeşitli Pascal örnek resimleri

1. txtall.pas (metin dosyaları açma)

(Ornekler\Pascal\okul\txtall.pas)

```
bat,txt,sys hangi dosya acilacak=
c:\nc.bat
Echo off
cd C:\utility\mouse
mouse
cd c:\utility\nc
nc
_
```

Resim 2 – Yazı dosyalarının gösteren program

```
program text_autoexec;
uses crt;
var dsy:text;hata:integer;satir:string[80];
    yol:string;
begin
  clrscr;
  writeln('bat,txt,sys hangi dosya acilacak=');
  readln(yol);
  assign (dsy,yol);
  {$i-} reset(dsy); {$i+}
  hata:=ioresult;
  if hata<>0 then
  begin
    writeln('dosya yok...');writeln('g/ç kodu=',hata:4);
    halt;
  end;

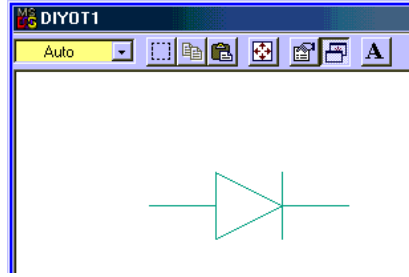
  while not eof(dsy) do
  begin
    readln(dsy,satir);
    writeln(satir);
  end;
  close(dsy);
  readln;
end.
```

Metin tabanlı dosyaları açar ve içeriğini ekrana listeler. Uzun dosyalarda hızla tümünü gösterip sadece son kısmı ekrana geliyor. Bunu geliştirerek “dir |more” komutundaki gibi bekletmeler ekleyebilirsiniz.

Pascal’ın yazım kuralları C diline çok benziyor. Satır sonlarında “;”, komutların birbirine benzer yapısı gibi. Fakat C diline göre Pascal daha kolay. Bir komut ile ilgili yardım almak isterseniz TP – Turbo Pascal veya BP – Borland Pascal editöründe komut üzerine imleci getirerek Ctrl+F1 tuşuna basınız.

2. diyot1.pas (grafik ekrana şekil çizme)

(Ornekler\Pascal\okul\diyot1.pas)



Resim 3 – Grafik ekranda diyot çizimi

Amaç grafik ekrana bir şekli çizmek. Bu programda uses kısmında kendi yaptığımız "grafik" adında unit kullanıyoruz. Yani bu program da iki parçadan oluşuyor. Eğer bu şekilde istemezseniz "grafik.pas" dosyasının içeri "diyot1.pas" içine kopyalar, uses kısmından da "grafik" unit'ini de çıkarırsınız. Ama bu sizin programınızın bir anda kocaman olmasını sağlar – şimdiki örnek için bu geçerli değil. Nereye ne yazdığınızı iyi biliyorsanız sorun yok. Compile – derleme sırasında zaten iki program otomatik olarak birleştiriliyor. Aslında uses kısmında ne varsa (crt, graph...) birleşiyor.

```
program diyot_graph;
uses crt,graph,grafik;
begin
ekran;setcolor(3);
moveto(100,100);lineto(150,100);
moveto(150,75);lineto(150,125);
lineto(200,100);lineto(150,75);
moveto(200,75);lineto(200,125);
moveto(200,100);lineto(250,100);
repeat until keypressed;
closegraph;
end.
```

Aşağıdaki grafik.pas "Compile" (F9 tuşu) edilince grafik.tpu haline geliyor. BGI Borland Graphic Interface – Borland grafik arabirimi anlamına geliyor. Grafik ekranınızın çözünürlüğü 640x450 pikseldir.

```
unit grafik;
interface
uses crt,dos,graph;
procedure ekran;
implementation
procedure ekran;
var graphdriver,graphmode:integer;
    err:integer;
begin
graphdriver:=detect;
initgraph(graphdriver,graphmode,'c:\bplunits'); { egavga.bgi dosyanız nerede? }
err:=graphresult;
if err<>gok then
begin
writeln('grafik hatası=',grapherrormsg(err));readln; halt;
end;
end;
begin
end.
```

Aynı diyotu yatay olarak kaydırmak için aşağıdaki programda grafik komutları olan Imagesize, Getimage ve Putimage kullanılmıştır. Eski görüntü xorput deyimi ile silinerek ilerlendiği için biraz kırılıyor. (Ornekler\Pascal\okul\diyot3.pas)

```
program diyot_graph; {hafıza kullanılarak kaydırılır}
```

```

uses crt,graph,grafik;
type nokta=record x,y,c:integer;end;
noktalar=array [1..10] of nokta;
const diyet:noktalar=((x:100;y:100;c:0),(x:150;y:100;c:1),
(x:150;y:75;c:0), (x:150;y:125;c:1),
(x:200;y:100;c:1),(x:150;y:75;c:1),
(x:200;y:75;c:0), (x:200;y:125;c:1),
(x:200;y:100;c:0),(x:250;y:100;c:1));
var i,j:integer;p:pointer;size:word;
begin
ekran;setcolor(4);setbkcolor(2);setcolor(blue);
for i:=1 to 10 do
begin
if diyet[i].c=0 then moveto(diyet[i].x,diyet[i].y)
else lineto(diyet[i].x,diyet[i].y);
end;
size:=imagesize(100,75,250,125);
getmem(p,size); getimage(100,75,250,125,p^);
putimage(100,75,p^,xorput);
for j:=1 to 300 do
begin putimage(100+j,75,p^,xorput);putimage(100+j,75,p^,xorput); end;
putimage(100+j,75,p^,xorput);
repeat until keypressed;
closegraph;
end.

```

Aşağıda da bir ok etrafında dönderiliyor: (Ornekler\Pascal\okul\ok.pas)

```

program ok_doner;
uses graph,grafik,crt;
type nokta=record x,y,c:integer; end;
noktalar=array[1..8] of nokta;
const ok:noktalar=((x:300;y:200;c:0),(x:330;y:230;c:1),(x:330;y:210;c:1),
(x:380;y:210;c:1),(x:380;y:190;c:1),(x:330;y:190;c:1),
(x:330;y:170;c:1),(x:300;y:200;c:1));
centerx=340; centery=200;
var i,degree:integer;rotate:noktalar;rad:real;

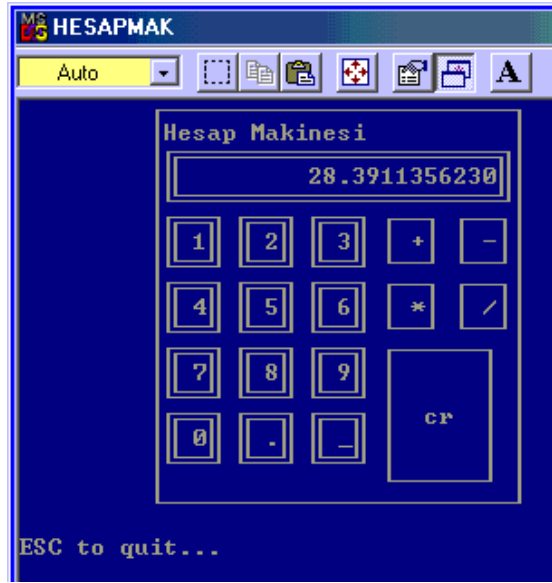
function dr(deg:integer):real;
begin dr:=pi*deg/180; end;

begin
ekran;
for degree:=0 to 360 do
begin for i:=1 to 8 do
begin rad:=dr(degree);
rotate[i].x:=round((ok[i].x-centerx)*cos(rad)-(ok[i].y-centery)*sin(rad))+centerx;
rotate[i].y:=round((ok[i].x-centerx)*sin(rad)+(ok[i].y-centery)*cos(rad))+centery;
rotate[i].c:=ok[i].c;
end;
setcolor(white);
for i:=1 to 8 do {cizim}
if rotate[i].c=0 then moveto(rotate[i].x,rotate[i].y)
else lineto(rotate[i].x,rotate[i].y);
delay(1500);
setcolor(black);
for i:=1 to 8 do {silim}
if rotate[i].c=0 then moveto(rotate[i].x,rotate[i].y)
else lineto(rotate[i].x,rotate[i].y);
end;
setcolor(white);
outtextxy(10,100,'Çıkmak için bir tus...');
repeat until keypressed;
closegraph;
end.

```


3. hesapmak.pas (hesap makinesi yapımı)

(Ornekler\Pascal\okul\hesapmak.pas)



Resim 4 – Hesap makinesi

Hesap makinesi örneği. Yine üniversitede uğraştığımız bir program. Hala bazı şeyler garip çalışıyor. Mesela sayıları telefondaki gibi sıralamışım, oysa klavyemizdeki rakamların sırası başka dikkat ederseniz. "." tuşunun yanındaki "_" alt çizginin bir fonksiyonu yok. Siz belki karekök, kare, 1/x benzeri bir komut ekleyebilirsiniz.

```

program hesap_mak;
uses crt;
type kututip=array[1..6] of char;
tip=record x1,y1,x2,y2,x3,y3:integer;yazim:string[15];
end;
kutu=array[1..19] of tip;

var tus:char;tusl:string[1];
sayi:string;sayi1,sayi2,sonuc:real;
islem:char;
hane:byte;hata,i:integer;
nokta:boolean;

const kutu1:kututip=(#196,#179,#218,#191,#192,#217);
      kutu3:kututip=(#205,#186,#201,#187,#200,#188);

kutucik:kutu=((x1:10;y1:1;x2:35;y2:19;x3:11;y3:2;yazim:'H
esap Makinesi'),
(x1:11;y1:3;x2:34;y2:5;x3:33;y3:4;yazim:'),
(x1:11;y1:6;x2:14;y2:8;x3:13;y3:7;yazim:'1'),
(x1:16;y1:6;x2:19;y2:8;x3:18;y3:7;yazim:'2'),
(x1:21;y1:6;x2:24;y2:8;x3:23;y3:7;yazim:'3'),
(x1:11;y1:9;x2:14;y2:11;x3:13;y3:10;yazim:'4'),
(x1:16;y1:9;x2:19;y2:11;x3:18;y3:10;yazim:'5'),
(x1:21;y1:9;x2:24;y2:11;x3:23;y3:10;yazim:'6'),
(x1:11;y1:12;x2:14;y2:14;x3:13;y3:13;yazim:'7'),
(x1:16;y1:12;x2:19;y2:14;x3:18;y3:13;yazim:'8'),
(x1:21;y1:12;x2:24;y2:14;x3:23;y3:13;yazim:'9'),
(x1:11;y1:15;x2:14;y2:17;x3:13;y3:16;yazim:'0'),
(x1:16;y1:15;x2:19;y2:17;x3:18;y3:16;yazim:'.'),
(x1:21;y1:15;x2:24;y2:17;x3:23;y3:16;yazim:'_'),
(x1:26;y1:6;x2:29;y2:8;x3:28;y3:7;yazim:'+'),
(x1:31;y1:6;x2:34;y2:8;x3:33;y3:7;yazim:'-'),

```

```

(x1:26;y1:9;x2:29;y2:11;x3:28;y3:10;yazim:')),
(x1:31;y1:9;x2:34;y2:11;x3:33;y3:10;yazim:'/''),

(x1:26;y1:12;x2:33;y2:18;x3:29;y3:15;yazim:'cr'));

procedure kutuciz(x1,y1,x2,y2:integer;kutu:kututip);
var i:byte;
begin
for i:=(x1+1) to x2-1 do
begin
gotoxy(i,y1);write(kutu[1]);
gotoxy(i,y2);write(kutu[1]);
end;{for}
for i:=(y1+1) to y2-1 do
begin
gotoxy(x1,i);write(kutu[2]);
gotoxy(x2,i);write(kutu[2]);
end;{for}
gotoxy(x1,y1);write(kutu[3]);
gotoxy(x2,y1);write(kutu[4]);
gotoxy(x1,y2);write(kutu[5]);
gotoxy(x2,y2);write(kutu[6]);

end;{pro-kutuciz}

procedure ses;
begin sound(720);delay(150);nosound; end;

procedure sonucyaz(son:real);
begin
gotoxy(13,4);write(' ');gotoxy(14,4);
write(son:20:10);
end;

```

```

procedure sinyal(x1,y1,x2,y2,tusdeg:integer);
begin
    kutuciz(x1,y1,x2,y2,kutu1);gotoxy(x2-1,y2-1);
    textbackground(red);write('
',tusdeg);textbackground(blue);
    sayi:=sayi+tusl;gotoxy(12,4);write(sayi:22);
    ses;delay(150);kutuciz(x1,y1,x2,y2,kutu3);
end;

procedure yaz(yazi:char);
begin
    inc(hane);
    if hane<13 then
        begin
            case tus of
                '1':sinyal(11,6,14,8,1);
                '2':sinyal(16,6,19,8,2);
                '3':sinyal(21,6,24,8,3);
                '4':sinyal(11,9,14,11,4);
                '5':sinyal(16,9,19,11,5);
                '6':sinyal(21,9,24,11,6);
                '7':sinyal(11,12,14,14,7);
                '8':sinyal(16,12,19,14,8);
                '9':sinyal(21,12,24,14,9);
                '0':sinyal(11,15,14,17,0);
            end;{case}
        end;{if}
    end;{pro-yaz}

begin
    textbackground(blue);clrscr;
    repeat
        begin
            gotoxy(1,21);write('ESC to quit...');
            gotoxy(36,4);write('
');
            for i:=1 to 19 do
                begin
                    if (i=1)or(i=19)or(i=18)or(i=17)or(i=16)or(i=15) then
                        kutuciz(kutucik[i].x1,kutucik[i].y1,kutucik[i].x2,kutucik[i].y2
                        ,kutu1)
                    else
                        kutuciz(kutucik[i].x1,kutucik[i].y1,kutucik[i].x2,kutucik[i].y2
                        ,kutu3);
                        gotoxy(kutucik[i].x3,kutucik[i].y3);
                        write(kutucik[i].yazim);
                end;{for}
            gotoxy(1,24); tus:=readkey;tusl:=tus;
            case tus of
                '0'..'9': yaz(tus);
                '.':begin
                    if nokta=false then begin
                        kutuciz(16,15,19,17,kutu1);gotoxy(18,16);
                        textbackground(red);write('.');textbackground(blue);

```

```

ses;sayi:=sayi+'.';nokta:=true;
kutuciz(16,15,19,17,kutu3);
end;{if}
end;{begin}
'+':begin
    kutuciz(26,6,29,8,kutu3);nokta:=false;ses;
    val(sayi,sayi1,hata);
    gotoxy(12,4);write('+');
    sayi:="";islem:="+";delay(150);hane:=0;
    sonucyaz(0);kutuciz(26,6,29,8,kutu1);
end;
'-':begin
    kutuciz(31,6,34,8,kutu3);nokta:=false;ses;
    val(sayi,sayi1,hata);
    gotoxy(12,4);write('-');
    sayi:="";islem:"-";delay(150);hane:=0;
    sonucyaz(0);kutuciz(31,6,34,8,kutu1);
end;
'*':begin
    kutuciz(26,9,29,11,kutu3);nokta:=false;ses;
    val(sayi,sayi1,hata);
    if sayi1=0 then exit;
    gotoxy(12,4);write('*');
    sayi:="";islem:="*";delay(150);hane:=0;
    sonucyaz(0);kutuciz(26,9,29,11,kutu1);
end;
'/':begin
    kutuciz(31,9,34,11,kutu3);nokta:=false;ses;
    val(sayi,sayi1,hata);
    if sayi1=0 then exit;
    gotoxy(12,4);write('/');
    sayi:="";islem:="/";delay(150);hane:=0;
    sonucyaz(0);kutuciz(31,9,34,11,kutu1);
end;
#13:begin
    kutuciz(26,12,33,18,kutu3);ses;delay(150);
    gotoxy(12,4);write(' ');
    val(sayi,sayi2,hata);sayi:="";
    kutuciz(26,12,33,18,kutu1);
    case islem of
        '+':sonuc:=sayi2+sayi1;
        '-':sonuc:=sayi1-sayi2;
        '*':sonuc:=sayi2*sayi1;
        '/':sonuc:=sayi1/sayi2;
    end;{case}
    gotoxy(12,4);write('
');gotoxy(22,4);write(sonuc:12:5);

sonucyaz(sonuc);nokta:=false;sayi:='0';sayi1:=0;sayi2:=0;
sonuc:=0;hane:=0;
end;{case-begin}
end;{CASE}
end;{repeat}
until tus=#27;
end.

```

4. matrice.pas (matrisler)

(Ornekler\Pascal\okul\matrice.pas)

```

matris elemanları girilir

matrisin [1,1] eleman11
matrisin [1,2] eleman12
matrisin [1,3] eleman15
matrisin [1,4] eleman18

matrisin [2,1] eleman19
matrisin [2,2] eleman16
matrisin [2,3] eleman15
matrisin [2,4] eleman14

matrisin [3,1] eleman18
matrisin [3,2] eleman18
matrisin [3,3] eleman18
matrisin [3,4] eleman18

matrisin [4,1] eleman17
matrisin [4,2] eleman17
matrisin [4,3] eleman16
matrisin [4,4] eleman14

a[1,1]= 1      a[1,2]= 2      a[1,3]= 5      a[1,4]= 8
a[2,1]= 9      a[2,2]= 6      a[2,3]= 5      a[2,4]= 4
a[3,1]= 8      a[3,2]= 8      a[3,3]= 8      a[3,4]= 8
a[4,1]= 7      a[4,2]= 7      a[4,3]= 6      a[4,4]= 4

aranılan elemani gir8
aranan sayı a[1,4] dır
aranan sayı a[3,1] dır
aranan sayı a[3,2] dır
aranan sayı a[3,3] dır
aranan sayı a[3,4] dır
8 sayisi toplam= 5 adet var
yerleri degişecek sayılar
1. sayının satırı=3
1. sayının sutunu=3
2. sayının satırı=1
2. sayının sutunu=1
A[3,3]sayı=8
A[1,1]sayı=1
A[3,3]sayı=1
A[1,1]sayı=8

```

Resim 5 – Matris veri giriş ve sonuçları

Sonuçta matematik ile programlama çok yakın konular. Trigonometri, matris ve benzeri konuları, grafik veya bilimsel işlemler içeren program üretirken kullanmamız gerekiyor. Basit bir çizginin etrafında dönderilmesi için sinüs ve kosinüs fonksiyonlarını kullanarak yapabiliriz. Delphi örneklerindeki saat projesine bir göz atınız. Ya da dizi ve matris ile aynı biçimdeki büyük veri yığınları ile kolayca uğraşabiliriz.

İki boyutlu olan a[1..4, 1..4] değişkenimiz için klavyeden 16 değer giriyoruz. Ekrana tekrar bu sayıları bize gösteriyor. Sonra sayı arama ve listeleme yapıyor. İki sayının değeri yer değiştirme – swap yapılarak program bitiyor. Yazıcı çıktısı verilen yer pasif halde, uses kısmındaki printer komutuna da dikkat.

```

program matrice;
uses crt,printer;
const sat=4;sut=4;
var i,j,k,l,asayı,top,yed:integer;
    a:array[1..sat,1..sut]of integer;

begin
  clrscr;
  writeln('matris elemanları girilir');
  for i:=1 to sat do
    begin
      writeln;

```

```

for j:=1 to sut do
begin
write('matrisin [' ,i,',',j,'] elemanı');
read(a[i,j]);
end; {for}
end;
readln;
clrscr;
{ekran çıkışı}
for i:=1 to sat do
begin
writeln;
for j:=1 to sut do
write(' ',a[ ,i,',',j,'] = ',a[i,j]);
end;{for}
{printer çıkışı}
for i:=1 to sat do
begin
writeln;
for j:=1 to sut do
write(lst,a[ ,i,j,']=',a[i,j]);
end;{for}
readln;
clrscr;
{eleman yeri bul}
write('aranılan elemanı gir');
readln(asayi);
top:=0;
for i:=1 to sat do
for j:=1 to sut do
begin
if a[i,j]=asayi then
begin
inc(top);
writeln('aranan sayı ',a[ ,i,',',j,'] dır');
end;{if}
end;{for}
writeln(asayi,' sayisi toplam= ',top:4,' adet var');
{eleman yer degiştir}
readln;
clrscr;
writeln('yerleri degişecek sayılar');
write('1. sayının satırı=');readln(i);
write('1. sayının sütunu=');readln(j);
write('2. sayının satırı=');readln(k);
write('2. sayının sütunu=');readln(l);
writeln('A[ ,i,',',j,']sayı=',a[i,j]);
writeln('A[ ,k,',',l,']sayı=',a[k,l]);
yed:=a[i,j];
a[i,j]:=a[k,l];
a[k,l]:=yed;
writeln('A[ ,i,',',j,']sayı=',a[i,j]);
writeln('A[ ,k,',',l,']sayı=',a[k,l]);
readln;
end.

```

Bölüm 5 – Delphi

İşte kitabın özü burası bence. Size faydalı olabilecek örnekleri açıklıyorum. Yarisından fazlası kendi projelerim.

DOS programcılığından farklı olarak artık sadece kod kısmı yok. İhtiyacımız olan yeni şey form üzerinde de neler olduğu. Bu bilgiler Delphi (*.dfm) ve VB'de (*.frm) ayrı dosyalarda saklanıyor. Bazı programlarda o kadar çok nesne var ki, bu bilgiler bazen sayfalarca sürüyor. Eğer CD'niz yok ise bu bilgilere bakarak programlarınızı hazırlayabilirsiniz.

Delphi'yi yada Visual Pascal da diyebiliriz, öğrenmek için Object Pascal dilini bilmeniz gerekiyor. Pascal temiz ve düzenli çalışmayı sevenlerin dilidir. Büyük küçük harfli kod kullanılması fark etmiyor ama okunaklı olması açısından yazarken ya büyük harf, ya da küçük harf kullanmanızı tavsiye ederim. Visual Basic, Delphi ve Visual C karşılaştırması yaparsak yazım ve okunaklılık olarak VB en basit, hatta çocuksu, VC ve Java ise karmaşık ve itici. Delphi tam programcı dostu. Delphi'nin yardım ve içindeki örnek programları VB ve VC kadar geniş değil ama programcıyı tökezlemeden rahat bir ortam sunuyor. Pencerelemlere ve aptalca arabirimlerle programcıyı boğmadan tasarım imkanı veriyor.

Object Repository – Nesne deposu, yani "File*New..." menüsünü incellerseniz application – uygulama, component – nesne, DLL, ActiveX Form, Thread Object, Web Server Application gibi zengin nesneleri görebilirsiniz.

Borland Database Engine yani BDE ile paradox, dbase, Access, FoxPro ve Excel gibi SQL* – Structured Query Language (Yapılandırılmış Sorgulama Dili)'ne dayalı ODBC (Open Database Connectivity) kaynaklarına erişebilirsiniz.

Delphi nesneleri yeterli olduğu halde, Visual Basic Custom Controls yani VBXs rahatlıkla Delphi içine "Component*Import ActiveX Control..." menüsünden entegre edilebilir.

Delphi IDE – Integrated Development Environment (Entegre Geliştirme Ortamı) ile hızlı ve rahat biçimde özgün programlar üretebilirsiniz. VCL – Visual Component Library (Görsel Nesne Kütüphanesi) yani nesnelerimiz kullanım amaçlarına göre ana menümüzün altında gruplandırılmışlardır. Standart, veritabanı, raporlama, internet ve diğer yardımcı konularda nesneler sıralanmıştır. Geriye uyumluluk için Win 3.1 ve Delphi1.0 uyumlu nesneleri vardır. Fakat artık Microsoft bile, Win 3.1 ve Win 95 işletim sistemlerine destek vermiyor.

SpeedBar – hızlı erişim çubuğundaki komutları Office'deki gibi özelleştirebilirsiniz. Mesela VB'deki gibi, "Run (F9)" düğmesinin yanında "Program Reset (Ctrl+F2)" da olsa ne iyi olur, değil mi? "Save All" düğmesinin yanına "Close All" da güzel yakışır. Eğer bileşen paletinizde tüm sayfalar gözüküyorsa tek yapabileceğiniz hile, bileşen paketlerinin adlarını kısaltmaktır. Standart yerine Std, Internet yerine Net, ActiveX yerine AX gibi...

Object Inspector – nesne özellikleri kısmından seçili olan nesne(ler)in özelliklerine - Properties ve olaylarına - Events erişebilirsiniz.

Kullanıcı arabiriminiz olan GUI – Graphical User Interface oluştururken, belli nesneleri sağa üste alta biştirmek, Grow to largest – en büyüğüne büyütme, Send to Back – geriye yollamak, Space equally – yatay dikey eşit aralık vermek kolaydır. Seçili nesneyi fare ile değil de klavyeden de konumlandırmak boyutlandırmak için Ctrl veya Shift ile yön tuşlarını deneyebilirsiniz.

Genellikle Delphi'yi ilk kurduğumda "Tools*Environment Options" diyerek şunları işaretlerim; Show compiler progress – Derleme işlemini gösterir, Minimize on run – Programımızı çalıştırınca Delphi görev çubuğuna küçülür, Show component captions – Nesne başlıklarını gösterir. Unit yani kod yazma yerine de sağ tıklayıp "Properties"; Undo after save – Kayıttan sonra geri al imkanı verir, Zoom to full screen – unitleri tam ekran görerek kod yazabilirsiniz, Color : Classic – yani bildiğimiz Borland Pascal veya Turbo Pascal renkleri, Code Insight Delay : 0.5 sec – kod tamamlaması 0.5 saniye gecikmeli.

Üzerine hiçbir şey koymasanız, hiç kod yazmasanız da projenizin *.exe dosyası çok büyük olabilir. Küçük olması için "Project*Options*Packages" e gelerek, "Build with runtime packages" deyin. Dosyanız 300 KB'tan 14 KB'a düşüverir. Fakat başka bilgisayarda çalışması için vcl30.bpl (1.4 MB), vcl40.bpl (1.8 MB), vcl50.bpl (1.9 MB) gibi sistem dosyalarının da orada olması gerekir. Yine "Project*Options" kısmında uygulamanızın simgesi, başlığı, yardım dosyası, hangi formun ana form olduğu gibi ayarlara ulaşabilirsiniz. Projeniz çok büyük de olsa compile - derleme işleminin ne kadar hızlı olduğunu görebilirsiniz. Oysa VB ve CBuilder'da bu süre daha uzundur.

* Ek 2 – SQL kısmına bakınız.

Kısayol tuşlarımız (Delphi 7'den üst versiyonlarda değişiklikler var):

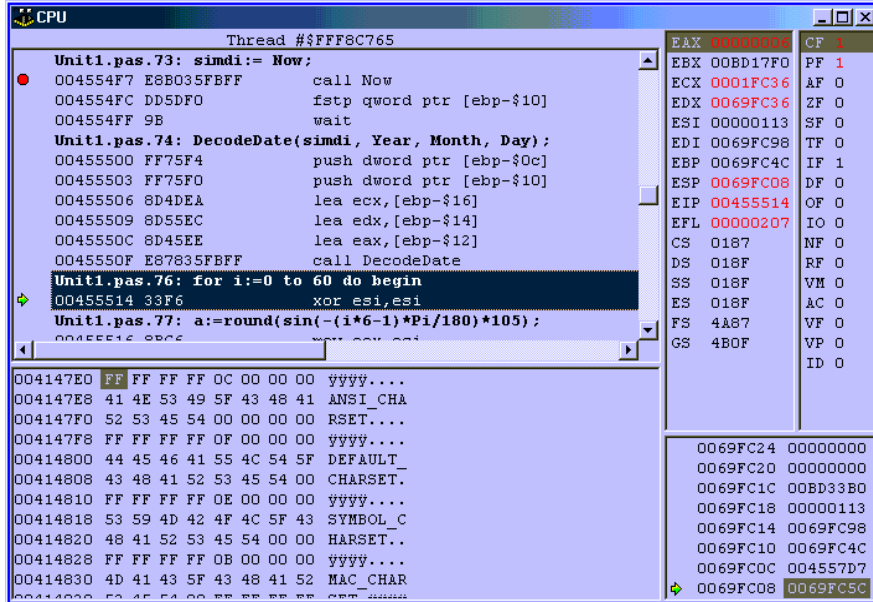
F9	: Çalıştır
Ctrl+F9	: Sadece hatalara bak ve derle...
Ctrl+F12	: Unit listemiz
Shift+F12	: Form listemiz
F11	: Object Inspector'u gösterir
F12	: Form ile unit arasında geçiş yapar (toggle yapmak)
F8 ve F7	: Projeyi adım adım çalıştır
Ctrl+F2	: Çalışan programı resetler yani kapatıp ve form tasarımına döner
Ctrl+S	: Aktfi penceredeki bilgileri kaydet
Ctrl+F ve R	: Bul (Find) veya yer (Replace) değiştir
F1	: İmleç neredeyse onunla ilgili konuda yardım ekrana gelir
Ctrl+Boşluk	: En çok kullandığım tuşlardan biri, kodu tamamlamaya zorlar
Ctrl+J	: Hazır deyim kalıplarını listeler
Ctrl+Y	: Pascal dilinden kalma satır sil komutu
Alt+F12	: Sürpriz, form nesneleriniz artık yazı biçiminde, ince iş yapanlara...

Tabii ki bir programın çalışmasını belli bir satırda durdurmak için F5 tuşu ile o satıra durak noktası koymamız gerekir. Eğer sadece belli durumda durak noktasının çalışmasını istersek "Edit breakpoint*Condition" ile şartımızı belirtebiliriz. F9 ile çalıştırdığımız program, kırmızı renkli bu durak noktasına gelince bizi bekler. O zaman ne yaparız? Delphi 3.0 ve sonrası için fly – by evaluation hints – fare ile bir değişkenin üzerine gelince ipucu olarak onu değerini görebilir, F8 ile çalışmaya satır satır devam ederken değişkenlerimizin değerinin nasıl değiştiğini görebiliriz. Canlı yayın yani... Bu şekilde bilgi tarama zor gelirse "Run* Add Watch..." (Ctrl+F5)" ile Watch List - seyretme listesine değişkenleri ekleyebilirsiniz. Klavyeden Insert tuşu ile yeni değişken ekleyebilir, Delete ile silebiliriz.

VB4 dizileri tamamen izlememize izin vermezken, VB5 ve üstünde ise bu seyretme işi diziler için çok hoş bir hale gelmiştir. Yani tüm dizideki elemanlara sadece dizi değişkeninin adını "Debug*Add Watch" komutu ile Watches kısmına eklemekle erişebilirsiniz. Ayrıca VB'de run time - çalışma durumunda immediate – anında penceresinde

? List1.ListCount + 5

gibi komut kullanılabilir. ?, Print komutu ile aynı anlamdadır. Örnek komut ekrana List1 nesnesinin liste eleman sayısına beş ekleyerek değerini yazar. Delphi'de de "Run*Evaluate/Modify..." komutu ile istediğiniz nesne değerine ulaşip matematiksel işlem yapabilirsiniz. "View*Debug Windows*CPU" programımızın disassembler edilmiş halini verir. Bu şekildeki programlamaya reverse engineering – ters mühendislik de denir.



Resim 1 – Saat projemizin işlemci içindeki hali

Sonuç olarak Watch işlemi kurt programcılarının eli koludur, çalıştığınız dilin böcek ayıklama adımlarını iyi öğreniniz.

Değişken tipleri integer (*intecır diye okunur*), real (*riıl diye okunur*), boolean (*buuliin diye okunur*), char (*çar diye okunur*), string - metin ve variant'dır. Kullandığınız her değişkeninizi var – variable (değişken) kısmına yazmanız, tanımlamanız gerekiyor.

Her dilde olduğu gibi yazım kuralı olarak Türkçe'de kullandığımız bazı kelimeleri kod yazmada deyim olarak kullanamayız: not, as, file, in, on, of, at, name, is, set, mod, var, nil...

Prosedür ve fonksiyonlarımızı Unit (*yunit diye okunur*) – ünite denilen editör kısmına yazarız. Bazı programlarda sadece nesnelerin birkaç özelliği ile oynayıp hiç kod yazmadan da program yapabilirsiniz. Fakat çoğu zaman böyle kolay olmuyor :) Pascal örneklerindeki diyot1.pas örneğinde kendi ünitemizi *.tpu olarak tanımlamayı görüyoruz. Delphi'deki unit yapısına çok benzer. Ama kullanım olarak farklıdır. Çünkü bunlar Delphi'de soyadı *.pas olan program kodlarımızın tutulduğu alandır. Pascal'da unit sonuna **Begin End.** yazmak gerekli iken, Delphi'de en sonda bir begin'i olmayan **End.** ile karşılaşıyoruz. **Begin End.** kısmı ile DLL oluştururken karşılaştıracğıız.

Programımıza her form eklediğimizde o formun komut kısmı Unit2, Unit3 şeklinde seri olarak oluşur. Birbirinden bağımsız olarak formların kodlarını yazarız, fakat amaç bir projede birden fazla program yazmak değildir. uses kısmı ile birbirlerinden haberdar olmalarını sağlarız. Programımızın en üst kısmındaki uses kısmına bazen el ile yazdığımız başka unit veya hazır kütüphaneleri eklememiz gerekebilir. Bunların sıraları önemli değildir. Artık örneğin Unit2'deki bir prosedürü unit1 içinde kullanabiliriz. Dikkat ederseniz artık Pascal'ın crt ünitesi Windows ile yer değiştirmiş.

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, inifiles, unit2, unit3;
```

Komut blokları C dilinde { ve } iken Pascal'da begin ve end'dir. C ile aynı olarak satır sonlarında ";" kullanılır. VB dilindeki *end if* veya *end case* deyimi Delphi'de yoktur. Blok belirten begin ve end yeterlidir. End'ler gittikçe kalabalıklaşınca en iyisi daha yazarken **end;** // if (a<... gibi açıklamalar eklemektir.

Bir nesneye kod yazmak için "NesneAdiOlayAdi()" gibi örneğin, Button1'in OnClick – tıklandığında olayı Button1Click() olarak oluşur. Olayın cinsine göre () parantez içindeki parametreler değişir. Dikkat ettiyseniz olayların başında Procedure yazıyor. Prosedüre değer yollanabilirken, onlardan değer dönemmez. Bunun için Pascal dili ile aynı olarak Function tanımlarız. Bu sayede sık kullanılacak olan kod bloklarını bir yere toplayıp, daha kısa ve hızlı program üretebiliriz. El ile yani manuel tanıttığımız fonksiyon ve prosedürleri örneklerde görebilirsiniz:

```
...
procedure TaraDir(fdirName:string;fext:string);
...
function IsPrime (N: LongInt): Boolean;
```

Delphi'de en sevdiğim fonksiyonlar StrToInt gibi çevrim yapanları. Fakat VB'deki **Left\$()**, **Right\$()** ve **Mid\$()** gibi hoş metin işlem komutları için Delphi'de biraz uğraşmak gerekiyor. Bir de "^" operatörü ile üs almak VB'de doğal iken Delphi'de math unitini ekleyip **power(x,y);** demek gerekiyor. VB için çok basit bir hareket olan formun ShowInTaskbar özelliği ile görev çubuğunda uygulamanızın gözükmemesi sağlanıyor. Bizim "Ornekler\Delphi\CD" 'deki CD örneğimizde bu aranan bir özellik...

Eğer Pascal'daki writeln(); gibi bir komut isterseniz, örneğin form için bir düşmenin tıklanma olayına şunları yazın:

```
Var
  F : TFont;
begin
  F:=TFont.Create;
  F.Name:='Tahoma';
  F.Style:=[fsBold];
  F.Size:=30;
  Canvas.Font.Assign(F);
  Canvas.Brush.Style := bsClear;
  Canvas.TextOut( 30, 60, 'Program 1.0');
  ...
```

x ve y değerleri ile oynayıp kayan yazı gibi bir çalışma yapabilirsiniz. Aynı şekilde VB’de de:

```

...
Fontname = "Arial"
FontBold = True
FontSize = 20
CurrentX = 700           'twip yani 1 inch = 1440 twip = 2,54 cm
CurrenY = 1400
Print "Program 1.0"
...

```

Boolean ifadeler Delphi’de True 1, False 0 değerindedir. VB ve OLE otomasyonu için ByteBool, WordBool ve LongBool veri tiplerinde True -1, False 0’dır. Üzerilerinde matematiksel işlem yapılabilir:

```

Caption:= IntToStr( Ord (1 < 6) + 4 );      // form başlığında 5 yazar
devam_mi:= not (sayi1=sayi2);              // karşılaştırma, := atama operatörüdür

```

VB’de ise mantıksal ifadelerde True -1, False 0 değerindedir. En azından C, Pascal ve Basic False’ın 0 olmasında ortaklar.

```

Caption = (1 < 6) + 4           ' form başlığına 3 yazar

```

Küme işlemleri de kullanışlı olabilir, **in** ile sayı değişkenimiz 1,5,20,23 dahil olarak 1’den 5’e veya 20’den 23’e kadar olan sınır içinde ise işlem yapılır:

```

if sayi in [1..5, 20..23] then toplam:=toplam+sayi;

```

Genellikle Pascal’da dosyalama konusunda karşılaştığımız tip tanımlamaları aşağıdaki şekilde yapılabilir. Kullanırken de değişkenler arasına nokta konularak **Tarih.Ay:=5;** gibi yapılır.

```

type
Tarih = record
  Yil: Integer;
  Ay: 1..12;
  Gun: 1..31;
end;

```

C dilindeki gibi variable typecast – değişken tipi atamaları vardır. Fakat bir türlü değişkenin tipine karar verememişseniz, yavaş çalışan Variant tipi ile değişken tanımlayabilirsiniz.

```

var karakter: char;
...
Shortint(karakter) := 122;           // karakter:='z'; eşdeğeri

```

Yine C dili ile benzer olarak break ve exit komutları Pascal’da da kullanılır. break ile for, while gibi döngülere devam edilir, exit ile prosedür veya fonksiyondan çıkılır.

Pointer - işaretçi tanımlama da şöyledir:

```

var    p: ^integer;
      x: integer;
begin
  p:= @x;           //x'in adresi
  x:= 100;
  if p <> nil then ShowMessage(inttostr(p^)); //aslinda ekrana x, 100 gelir
...

```

Format fonksiyonu çok güçlü metin biçimleme yardımcımızdır. C dilindeki printf benzeri, d tamsayı, h onaltılık, s metin, e üstel, f kayar noktalı anlamındadır. %-8d yazımının anlamı sola dayalı rakamdır. Onluktan onaltılığa veya onaltılıktan onluk sisteme çevirmek için bu komut kullanılabilir.

```

caption := Format( 'İlk %3d, son %3d', [ n1 , n2 ] );

```

Nesne özellikleri küme işlemleri gibi kullanılır:


```

Memo1.Font.Style := Memo1.Font.Style - [fsBold] + [ fsItalic , fsUnderline ];
Button1.Font := Button2.Font;           // font kopyalama
Label1.Alignment := taRightJustify;

```

Eğer bir uygulama projemiz varsa projemiz ile ilgili genel bir nesne adımız var: **Application**

```

...
Caption := Application.Title;           // uygulamanın başlığı yazılır
Caption := ExtractFilePath(Application.ExeName); // bulunduğu yol yazılır
...

```

VB’de ise: **App**

```

...
Caption = App.EXENAME
Caption = App.Path
...

```

Delphi’de hata olması muhtemel yerlere try (*tray diye okunur*) – dene komutu ile güvence verebilirsiniz.

```

...
try
    Memo1.Font.Name:=’Arial Tur’;
    Memo1.Font.Size:=18;
except
    on Exception do ShowMessage(’Fontta bir sorun var!’);
end;
...

```

Veri girişi için sadece basit bir InputQuery veya InputBox kullanmak daha faydalı:

```

var kelime:string;
begin
    if inputquery (’Aranan Kayıt’, ’Kelime giriniz:’, kelime) then
        begin
            table1.locate(’KELIME’,kelime,[loCaseInsensitive, loPartialKey]);
        end;
end;

```

Meraba.scr’dekinden farklı olarak programımızın bir kere çalışmasını proje dosyamıza şu kodları ekleyerek yapabiliriz:

```

program Tekcopy;

uses
    Forms,Dialogs,Windows,
    Unit1 in 'Unit1.pas' {Form1};

{$R *.RES}

var
    Hwnd: THandle;

begin
    Hwnd := FindWindow ('TForm1', 'Başlık');
    if Hwnd = 0 then
        begin
            Application.CreateForm(TForm1, Form1);
            Application.Run;
        end
    else
        begin
            MessageDlg ('Sadece 1 kere çalıştırılabilir...',mtInformation, [mbOk], 0);
            SetForegroundWindow (Hwnd);
        end
    end.

```

Bir nesneyi (Button, Edit,...) basitçe sürüklemek için şu kısa kod yeterlidir:

```
Procedure TForm1.Button1MouseDown(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
const sc_DragMove=$F012;
begin
    ReleaseCapture;
    Button1.Perform(WM_SYSCOMMAND, sc_DragMove, 0);
end;
```

Görev çubuğundan programın simgesini kaldırmak için:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
    SetWindowLong(Application.Handle, GWL_EXSTYLE, WS_EX_TOOLWINDOW);
end;
```

Farenizi istediğiniz yere yönlendirebilirsiniz, bu sayede fareyi belli koordinata da hapsedebilirsiniz:

```
procedure TForm1.Button1Click(Sender: TObject);
var nokta:TPoint;
begin
    GetCursorPos(nokta);           //farenin o anki koordinatı alınır
    SetCursorPos(nokta.x+10,nokta.y+10); //10 piksel sağ aşağı çapraz hareket ettirilir
end;
```

Fare tıklamasının taklit edilmesi, yani başka bir nesne – Edit1 tıklanınca sanki düğme – Button2 tıklanmış gibi yapmak:

```
procedure TForm1.Edit1Click(Sender: TObject);
begin
    SendMessage(button2.handle, WM_LBUTTONDOWN, 0, 0);
    SendMessage(button2.handle, WM_LBUTTONUP, 0, 0);
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
    MessageBeep(500);           //bip sesi
end;
```

Standart dizin seçme diyalog kutusu şu şekilde kullanılabilir:

```
uses FileCtrl;           // uses da neresi demeyin.
procedure TForm1.Button1Click(Sender: TObject);
var
    Result: integer;
    Dir: string;
begin
    Dir := 'C:\';
    if SelectDirectory(Dir, [sdAllowCreate, sdPerformCreate, sdPrompt], Result) then
        form1.Caption := Dir;
end;
```

Delphi'de yazılabilecek en küçük programı (*.dpr) proje dosyasına yazabilirsiniz:

```
program Beep;
uses
    Windows;
begin
    MessageBeep (0);
end.
```

Uygulamanızda birçok minik resim, simge gibi bitmap kaynaklar ve sabit metin dizisi varsa; resimleri Image Editor programı ile *.res resource haline getirerek, metinleri de not defterinde kendiniz yazarak "Delphi\Bin" dizinindeki brcc32.exe ile derleyerek daha hızlı kullanabilirsiniz. Harici dosyaları kullanıcılara ayrıca vermeniz

gerekirken, *.res kullanarak *.exe içine dahil etmiş olursunuz. TicTocToe örneğimizde sadece üç resim olduğu için bu yöntemi kullanmadım.

Grafik için:

```
var
  Form1: TForm1;
  Bmp: TBitmap;
implementation
{$R *.DFM}
{$R BITMAPS.RES}           //kaynak dosyamız (brcc32 ile elde edilen)
...
  Bmp.Handle := LoadBitmap (HInstance, 'bitmap dosya adı');
  DrawGrid1.Canvas.Draw (Rect.Left, Rect.Top, Bmp);
...
```

Metin için:

```
STRINGTABLE
BEGIN
  1, "Merhaba"
  2, "String Tablo"
  11, "İlk mesaj"
  12, "İkinci mesaj"
  13, "Üçüncü mesaj"
...
  20, "Onuncu mesaj"
END
```

```
implementation
{$R *.DFM}
{$R tablo.res}           //kaynak dosyamız (brcc32 ile elde edilen)
...
  string_mesaj := LoadStr(1);           //Merhaba
  ShowMessage (string_mesaj);
...
  string_mesaj := LoadStr(11);          //İlk mesaj
  ShowMessage (string_mesaj);
...
```

Not defteri ile bu yazıları tablo.rc olarak kaydediniz. Komut satırından "brcc32.exe -r tablo.rc" diyerek tablo.res dosyasını elde ediniz. Sabit metin veri yığını örneğin, menülerimizin dilini İngilizce, Türkçe ve Almanca gibi farklı dillerde kullanılabilmesi için oluşturabiliriz. GWBasic dilindeki satır numaraları gibi örneğin 1000 sonrası Türkçe, 2000 sonrası İngilizce gibi ayarlayabiliriz.

Casio FX-880P'de Basic ile yaptığım kağıt oyununda bilgileri şu şekilde program içine kaydedip kullanmıştım:

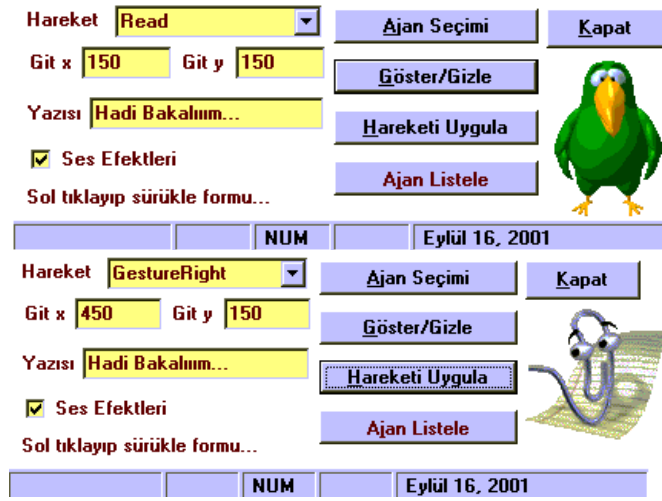
```
10  cls
20  restore           data ilklenir
30  dim kagit$(52)
40  data ak,kk,qk,jk,tk,9k,8k,7k,6k,5k,4k,3k,2k ...   diger kagit bilgileri
50  for i=1 to 52 : read kagit$(i) : next
```

data komutu ile yazdığım sabit bilgileri, read komutu ile kagit\$ adlı diziye aktarıyordum. Bir Hex editör ile *.exe dosyanın içindeki bu bilgilere erişilerek, mesela kullanıcıların bilmemesi gereken bilgiler görülebilir. Tavsiyem bu tip veriyi şifrelemenizdir. Böylece 52 satırlık kod yazmaktan kurtulmuştum. Aslında kağıt bilgilerini Casio'nun Memory - yazı yazma kısmında dosya haline getirip Basic kısmından bunları okumayı denemiştım, ama bunu denerken karşıma Peek ve Poke komutları çıktı. Sürpriz olarak bu sırada tüm bilgileri silen program da yazmış ve çalıştırmıştım(?!). Tabii yedekleme diye bir şey olmadığı için yaptığım tüm emeklerim silinmişti. "Ornekler\Basic\peek.bas" örneğini inceleyiniz.

```
print peek (&h300)   '300H adresi bilgilerini ekrana yazar
poke &h300,254       '300H adresine 255 yazar
```

1. Agent (Microsoft Agent kullanımı)

(Ornekler\Delphi\Agent\Project1.dpr)



Resim 2 – Microsoft Agent kullanımı

Deneye deneye birçok özellikleri buldum. İnternet'ten araştırarak eklentiler yapılabilir. Microsoft'un sitesi olan MSDN kütüphanesi yeterince geniş.

Office 2000 ve üstü ile birlikte kurulan Microsoft Agent "Windows\MsAgent" klasörüne yerleşiyor. Robby, Peedy, Genie ve Merlin gibi diğer karakterleri yüklerseniz "Windows\MsAgent\Chars" dizinine bunların yerleştiğini görürsünüz. microsoft.com'dan da MSAgent yükleyebilirsiniz. CD'de "Diger\Extras" dizininde MsAgent.exe, peedy.exe, genie.exe gibi dosyaları bulabilirsiniz. Office 2000 veya XP'nin karakterleri olan F1, Rocky, Cin de kullanılabilirken, Office 97 karakterleri uyumsuz. "Ajan Seçimi" düğmemiz ile "Windows\MsAgent\Chars" dizinindeki veya başka bir yerdeki *.acs karakterleri açabiliriz. Office ajanlarını açabildiğimiz halde onların hareket isimleri tam buradaki gibi değil.

Ajanları Office VBA dili ile, web sayfası içinde de VBScript veya JavaScript dilleri ile çağırabilir ve kullanabiliriz. ActiveX ile ilgili Diğer Diller adlı kısımda açıklamalar bulabilirsiniz. Bilgisayarımızda MSAgent'in kurulu olup olmadığını "Ornekler\Delphi\Agent\kurulu_mu.html" adlı dosyayı çalıştırarak öğrenebilirsiniz:

```
<SCRIPT LANGUAGE=VBSCRIPT>

If HaveAgent() Then
    document.write "<H2 align=center>MS Agent Kurulu...</H2>"
Else
    document.write "<H2 align=center>MS Agent Bulunamadı...</H2>"
End If

Function HaveAgent()
    Dim agent
    HaveAgent = False
    On Error Resume Next
    Set agent = CreateObject("Agent.Control.1")
    HaveAgent = IsObject(agent)
End Function

</SCRIPT>
```

Delphi'de Agent kullanabilmek için "Component*Import ActiveX Control..." dedikten sonra "Microsoft Agent Control (Version 2.0)" gibi ifadeyi bulun, Palette page'i ActiveX seçerek Install deyin. Şapkalı bir vatandaşın nesnelerinize eklendiğini göreceksiniz. VB için de aynı şekilde ToolBox içine dahil edilebilir. Delphi örneği sonundaki VB örneğinde birden fazla karakterin de kullanılabilirdiğini görebilirsiniz.

Form üzerinde şunlar var:

```

object Form1: TForm1
  Left = 271
  Top = 104
  ActiveControl = Button2
  BorderStyle = bsNone
  Caption = 'Microsoft Agent Kullanımı'
  ClientHeight = 161
  ClientWidth = 416
  Color = clWhite
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clMaroon
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = [fsBold]
  OldCreateOrder = True
  Position = poScreenCenter
  ShowHint = True
  OnCreate = FormCreate
  OnMouseDown = FormMouseDown
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
    Left = 10
    Top = 35
    Width = 27
    Height = 13
    Caption = 'Git x'
  end
  object Label2: TLabel
    Left = 106
    Top = 35
    Width = 27
    Height = 13
    Caption = 'Git y'
  end
  object Label3: TLabel
    Left = 8
    Top = 8
    Width = 46
    Height = 13
    Caption = 'Hareket'
  end
  object Label4: TLabel
    Left = 9
    Top = 67
    Width = 34
    Height = 13
    Caption = 'Yazısı'
  end
  object Label5: TLabel
    Left = 8
    Top = 120
    Width = 155
    Height = 13
    Caption = 'Sol tıklayıp sürükley formu...'
  end
  object SpeedButton1: TSpeedButton
    Left = 202
    Top = 104
    Width = 129
    Height = 22
    Caption = 'A&jan Listele'
    OnClick = SpeedButton1Click
  end
  object Button1: TButton

```

```

    Left = 202
    Top = 72
    Width = 129
    Height = 22
    Caption = '&Hareketi Uygula'
    TabOrder = 0
    OnClick = Button1Click
  end
  object Agent1: TAgent
    Left = 368
    Top = 56
    Width = 32
    Height = 32
    Hint = 'asdasd'
    ControlData = {000300004F0300004F030000}
  end
  object ComboBox1: TComboBox
    Left = 64
    Top = 6
    Width = 131
    Height = 21
    ItemHeight = 13
    TabOrder = 2
    Text = 'Read'
    Items.Strings = (
      'Greet'
      'Read'
      'Write'
      'Think'
      'Confused'
      'Announce'
      'Search'
      'Show'
      'Hide'
      'Sad'
      'Pleased'
      'GestureRight'
      'GestureLeft')
  end
  object Edit1: TEdit
    Left = 43
    Top = 32
    Width = 57
    Height = 21
    TabOrder = 3
    Text = '150'
  end
  object Edit2: TEdit
    Left = 140
    Top = 32
    Width = 57
    Height = 21
    TabOrder = 4
    Text = '150'
  end
  object Edit3: TEdit
    Left = 48
    Top = 64
    Width = 148
    Height = 21
    TabOrder = 5
    Text = 'Hadi Bakalım...'
  end
  object Button2: TButton
    Left = 336

```

```

Top = 8
Width = 75
Height = 25
Caption = '&Kapat'
TabOrder = 6
OnClick = Button2Click
end
object Button3: TButton
Left = 202
Top = 40
Width = 129
Height = 22
Caption = '&Göster/Gizle'
TabOrder = 7
OnClick = Button3Click
end
object CheckBox1: TCheckBox
Left = 11
Top = 95
Width = 137
Height = 17
Caption = 'Ses Eftleri'
Checked = True
State = cbChecked
TabOrder = 8
OnClick = CheckBox1Click
end
object StatusBar1: TStatusBar
Left = 0
Top = 142
Width = 416
Height = 19
Panels = <
item
Width = 100
end
item
Width = 50
end

```

```

item
Width = 50
end
item
Width = 50
end
item
Width = 100
end>
SimplePanel = False
end
object Button4: TButton
Left = 202
Top = 8
Width = 129
Height = 22
Caption = '&Ajan Seçimi'
TabOrder = 10
OnClick = Button4Click
end
object Timer1: TTimer
Interval = 150
OnTimer = Timer1Timer
Left = 328
Top = 56
end
object OpenFileDialog1: TOpenDialog
FileName = '*.acs'
Filter = 'Microsoft Agent Karakteri (*.acs)|*.acs|Tüm
Dosyalar (*.*)|*.*'
InitialDir = 'c:\windows\msagent\chars'
Options = [ofReadOnly, ofHideReadOnly,
ofEnableSizing]
Title = 'Ajan Seçiniz'
Left = 368
Top = 96
end
end

```

Kod kısmı da şunlar:

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls, OleCtrls, AgentObjects_TLB, Spin, ComCtrls, ExtCtrls, Buttons;
```

```
type
```

```

TForm1 = class(TForm)
Agent1: TAgent;
Button1: TButton;
Button2: TButton;
Button3: TButton;
Button4: TButton;
CheckBox1: TCheckBox;
ComboBox1: TComboBox;
Edit1: TEdit;
Edit2: TEdit;
Edit3: TEdit;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;

```

```

Label5: TLabel;
OpenDialog1: TOpenDialog;
SpeedButton1: TSpeedButton;
StatusBar1: TStatusBar;
Timer1: TTimer;
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Timer1Timer(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  c:widestring;d:OleVariant;
implementation
{$R *.DFM}

procedure TForm1.FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
const
  SC_DragMove = $F012;
begin
  Begin
    ReleaseCapture;
    FORM1.perform(WM_SysCommand, SC_DragMove, 0);
  End;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
const
  CapPanel = 1; NumPanel = 2;
  ScrlPanel = 3; DatePanel = 4;
begin
  with StatusBar1 do
  begin
    if GetKeyState(VK_CAPITAL) <> 0 then
      StatusPanels[CapPanel].Text := ' CAP'
    else
      StatusPanels[CapPanel].Text := '';
    if GetKeyState(VK_NUMLOCK) <> 0 then
      StatusPanels[NumPanel].Text := ' NUM'
    else
      StatusPanels[NumPanel].Text := '';
    if GetKeyState(VK_SCROLL) <> 0 then
      StatusPanels[ScrlPanel].Text := ' SCRL'
    else
      StatusPanels[ScrlPanel].Text := '';
    StatusPanels[DatePanel].Text := FormatDateTime(' mmmm D, YYYY', now);
  end;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  c:='Merlin';
  agent1.Characters.Load(c, 'merlin.ACS');
  Agent1.Characters.Item[c].Set_AutoPopupMenu(True);

```

```

Application.HintColor:=claqua;
end;

procedure TForm1.Button1Click(Sender: TObject);
var a :OleVariant;
begin
agent1.Characters.Item[c].MoveTo(strtoint(edit1.text),strtoint(edit2.text),a);
agent1.Characters.Item[c].Play(ComboBox1.Text);
agent1.Characters.Item[c].Think(edit3.text);
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
Agent1.Characters.Item[c].ShowPopupMenu
(left+button3.left,top+button3.top+button3.height);
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
agent1.Characters.Unload(c);
close;
end;

procedure TForm1.CheckBox1Click(Sender: TObject);
begin agent1.Characters.Item[c].Set_SoundEffectsOn(CheckBox1.Checked); end;

procedure TForm1.Button4Click(Sender: TObject);
var uzanti:string;i:byte;dosyaadi:string;
begin
OpenDialog1.FileName:="";
OpenDialog1.Execute;
if OpenDialog1.FileName<>" then
begin
dosyaadi:=lowercase(OpenDialog1.FileName);
agent1.Characters.Unload(c);
uzanti:="";c:="";
i:=strlen(pchar(OpenDialog1.FileName));
while (dosyaadi[i]<>'.' ) do dec(i);           //uzanti koparilir
dec(i);                                         //nokta gidiyor
while (dosyaadi[i]<>'\'') do begin              //sadece adı lazim
uzanti:=dosyaadi[i]+uzanti;
dec(i);
end;
c:=uzanti;
agent1.Characters.Load(c, OpenDialog1.FileName);
end;
end;

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin agent1.ShowDefaultCharacterProperties; end;

end.

```

Aslında sadece Agent değil, form sürükleme yapıp, durum çubuğunda Num Lock, Caps Lock ve Scroll Lock gibi bilgileri de gösterdik. Formu sol tıklayıp sürükleme imkanı da veriyoruz.

Visual Basic dilinde de animasyon hazırlamak kolaydır. Peşpeşe komutlarla sanki bir olayı anlatıyorlarmış görüntüsü verebiliriz. Fıkra anlatmak veya bir şeyin tanıtımı gibi... Artık eksikler; seslendirme yani konuşmaları ve bizim yazdığımız veya mikrofondan konuştuğumuz kelimeleri sentezlemeleri. Eğer MSDN Library kütüphane sitesini incellerseniz, orada speech – konuşma ile ilgili Microsoft Voice Commands yeteneği, ayrıca 18 MB'lık MS Agent Software Development Kit programını indirebilir, kendi karakterinizi yapabilme gibi imkanlara kavuşursunuz. Şu anki kullanımımız bilgilendirme dışında bir işimize yaramıyorlar. (Ornekler\VB\MSAgent\Project1.vbp)

VB içinde tasarlarken bir liste kutusu, dört düğme, üç metin kutusu, üç etiket ve bir agent nesnesi kullanıyoruz. Merlin, Robby ve Genie'yi kurmayı unutmayın.


```

Dim GenieRequest
Dim RobbyRequest
Dim merlinRequest
Dim Genie
Dim Robby
Dim merlin

Private Sub Command1_Click() 'Fıkra
    Genie.MoveTo 100, 100
    Genie.Show
    Robby.MoveTo 250, 100
    Robby.Show
    Genie.Play "Announce"
    Set GenieRequest = Genie.Speak("Tavuk neden karşıya
geçti?")
    Robby.Wait GenieRequest
    Robby.Play "Confused"
    Set RobbyRequest = Robby.Speak("Ne biliyim. Sence
neden geçti?")
    Genie.Wait RobbyRequest
    Genie.Play "Pleased"
    Set GenieRequest = Genie.Speak("Karşı tarafta olmak
için...")
    Robby.Wait GenieRequest
    Robby.Play "Sad"
    Robby.Speak "Hiç sormamalıydım!"
End Sub

Private Sub Command2_Click() 'Cin'in hareket
listesi
    List1.Clear
    For Each Animation In Genie.AnimationNames
        List1.AddItem (Animation)
    Next
End Sub

Private Sub Command3_Click() 'Merlin'in şovu
    Dim strName As String
    merlin.MoveTo 400, 100
    merlin.Show
    merlin.Play "Greet"
    merlin.Play "Restpose"
    merlin.Speak "Merhaba!"
    merlin.Play "Announce"
    merlin.Speak "Ben Merlin."
    merlin.Play "Pleased"
    merlin.Speak "Tanıştığımıza memnun oldum!"
    merlin.MoveTo 250, 500
    merlin.Speak "Size bazı şeyler göstereceğim..."
    merlin.Play "Hide"
    merlin.Play "Show"
    merlin.Speak "bazı hareketlerimi..."
    merlin.Speak "İşte bir sihir numaram!"
    merlin.Play "DoMagic1"

```

```

merlin.Play "DoMagic2"
merlin.Play "Pleased"
merlin.Play "Restpose"
merlin.Play "Suggest"
merlin.Speak "Şimdi bir önerim var!"
merlin.Speak "İşte lamba ile bunu belirtirim..."
merlin.Play "Pleased"
merlin.Play "Restpose"
merlin.Speak "Okuyabilir..."
merlin.Play "Read"
merlin.Speak "ve yazabilirim."
merlin.Play "Write"
merlin.Play "Pleased"
merlin.Play "Restpose"
merlin.Speak "Teşekkürler"
merlin.Play "Wave"
merlin.Speak "Güle güle..."
merlin.Hide
End Sub

Private Sub Command4_Click() 'Kapat
End
End Sub

Private Sub Form_Load() 'İlk açılış
    Agent1.Characters.Load "Genie", "genie.acs"
    Agent1.Characters.Load "Robby", "robbie.acs"
    Agent1.Characters.Load "Merlin", "merlin.acs"

    Set Genie = Agent1.Characters("Genie")
    Set Robby = Agent1.Characters("Robby")
    Set merlin = Agent1.Characters("Merlin")

    Genie.Get "State", "Showing"
    Robby.Get "State", "Showing"

    Genie.Get "Animation", "Announce, AnnounceReturn,
Pleased,PleasedReturn "

    Robby.Get "Animation", "Confused, ConfusedReturn,
Sad, SadReturn"

    Set Genie = Agent1.Characters("Genie")
    Set Robby = Agent1.Characters("Robby")
End Sub

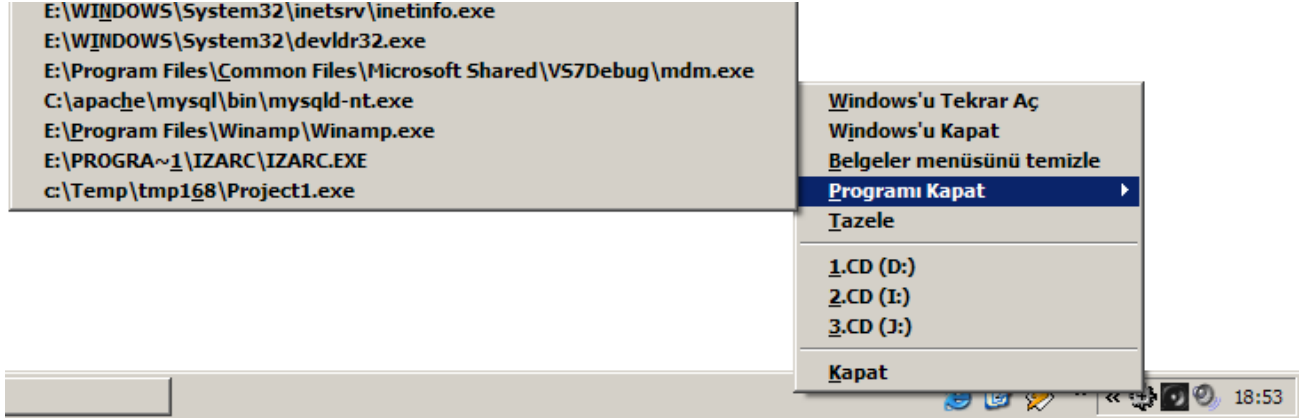
Private Sub List1_Click() 'Cin'e komut yollamak
    Genie.Stop
    Genie.MoveTo Text1.Text, Text2.Text
    Genie.Show
    Genie.Play List1.Text
    Genie.Speak Text4.Text
End Sub

```


2. CD (görev çubuğunda program çalıştırma)

(Ornekler\Delphi\CD\Project1.dpr)

Bu programda görev çubuğuna - taskbar simge ekleme, sürücüden CD - compact disk çıkarma, geri takma, Windows'u kapatma, yeniden başlatma komutlarını görüyoruz. Takma çıkarma işlemi DVD - digital video disk için de geçerli. En fazla 4 adet olabilen CD'leri otomatik olarak bulup, resimdeki gibi gösteriyoruz.



Resim 4 - CD programı

Proje dosyamız:

```
program Project1;
```

```
uses
  formkapa in 'formkapa.PAS',
  Forms, Windows,
  Unit1 in 'Unit1.pas' {Form1};
```

```
{ $R *.RES }
```

```
begin
  Application.ShowMainForm := False;
  IsLibrary := false;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
```

Görev çubuğundaki gereksiz program simgeyi yok etmek için yazacağımız unit:

```
unit formkapa;
```

```
interface
```

```
implementation
```

```
initialization
  IsLibrary := True;
```

```
end.
```

Kodlarımız:

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs, Menus, MMSystem,
StdCtrls, ShellAPI, ExtCtrls, ShlObj, Psapi, tlhelp32;
```

```
const
  wm_IconMessage = wm_User;
```

```

type
  TForm1 = class(TForm)
    PopupMenu1: TPopupMenu;
    N1CDA1: TMenuItem;
    N2CDA1: TMenuItem;
    Kapat1: TMenuItem;
    WindowsuTekrarA1: TMenuItem;
    N1: TMenuItem;
    WindowsuKapat1: TMenuItem;
    N2: TMenuItem;
    Timer1: TTimer;
    N2CDA2: TMenuItem;
    N2CDA3: TMenuItem;
    Belgelermensntemizle1: TMenuItem;
    ProgramKapat1: TMenuItem;
    azele1: TMenuItem;
    procedure Kapat1Click(Sender: TObject);
    procedure N1CDA1Click(Sender: TObject);
    procedure N2CDA1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormClick(Sender: TObject);
    procedure WindowsuTekrarA1Click(Sender: TObject);
    procedure WindowsuKapat1Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure N2CDA2Click(Sender: TObject);
    procedure N2CDA3Click(Sender: TObject);
    procedure Belgelermensntemizle1Click(Sender:
TObject);
    procedure tikla(Sender: TObject);
    procedure azele1Click(Sender: TObject);

  public
    procedure IconTray (var Msg: TMessage);
    message wm_IconMessage;
  private
    nid:TNotifyIconData;

  end;

var
  Form1: TForm1;
  ac1:boolean=true;
  ac2:boolean=true;
  ac3:boolean=true;
  ac4:boolean=true;
  cdler:array [0..4] of string;
  cdsayisi:byte=0;

implementation

{$R *.DFM}

procedure CreateWin9xProcessList(List: TstringList);
var
  hSnapshot: THandle;
  ProcInfo: TProcessEntry32;
begin
  if List = nil then Exit;
  hSnapshot :=
CreateToolHelp32Snapshot(TH32CS_SNAPPROCESS, 0);
  if (hSnapshot <> THandle(-1)) then
  begin
    ProcInfo.dwSize := SizeOf(ProcInfo);
    if (Process32First(hSnapshot, ProcInfo)) then
      begin

```

```

        List.Add(ProcInfo.szExeFile);
        while (Process32Next(hSnapshot, ProcInfo)) do
          List.Add(ProcInfo.szExeFile);
        end;
        CloseHandle(hSnapshot);
      end;
    end;
  end;

procedure CreateWinNTProcessList(List: TstringList);
var
  PIDArray: array [0..1023] of DWORD;
  cb: DWORD;
  I: Integer;
  ProcCount: Integer;
  hMod: HMODULE;
  hProcess: THandle;
  ModuleName: array [0..300] of Char;
begin
  if List = nil then Exit;
  EnumProcesses(@PIDArray, SizeOf(PIDArray), cb);
  ProcCount := cb div SizeOf(DWORD);
  for I := 0 to ProcCount - 1 do
  begin
    hProcess :=
OpenProcess(PROCESS_QUERY_INFORMATION or
PROCESS_VM_READ,
False,
PIDArray[I]);
    if (hProcess <> 0) then
    begin
      EnumProcessModules(hProcess, @hMod,
SizeOf(hMod), cb);
      GetModuleFileNameEx(hProcess, hMod, ModuleName,
SizeOf(ModuleName));
      List.Add(ModuleName);
      CloseHandle(hProcess);
    end;
  end;
end;

procedure GetProcessList(var List: TstringList);
var
  ovi: TOSVersionInfo;
begin
  if List = nil then Exit;
  ovi.dwOSVersionInfoSize := SizeOf(TOSVersionInfo);
  GetVersionEx(ovi);
  case ovi.dwPlatformId of
    VER_PLATFORM_WIN32_WINDOWS:
      CreateWin9xProcessList(List);
    VER_PLATFORM_WIN32_NT:
      CreateWinNTProcessList(List);
  end
end;

procedure CDlistesi;
var buffer : array[0..500] of char;
    temp : PChar;
    typ,i,ara : integer;
    MyProcList: TstringList;
    itme: TMenuItem;
begin
  cdsayisi:=0;
  GetLogicalDriveStrings( sizeof( buffer ), buffer );
  temp := buffer;
  while temp[0] <> #0 do

```

```

begin
  typ := GetDriveType( temp );
  if typ = DRIVE_CDROM then
    begin
      cdler[cdsayisi]:= copy(temp,0,1);
      inc(cdsayisi);
      end;
    temp := StrEnd( temp ) + 1;
  end;
  //ShowMessage(inttostr(cdsayisi));
  //ShowMessage(cdler[0]+'.'+cdler[1]+'.' +cdler[2]);
  if (inttostr(cdsayisi) = '0') then form1.N1.Visible:=false;
  if (cdsayisi>=1) then form1.N1CDA1.Visible:=true;
  if (cdsayisi>=2) then form1.N2CDA1.Visible:=true;
  if (cdsayisi>=3) then form1.N2CDA2.Visible:=true;
  if (cdsayisi>=4) then form1.N2CDA3.Visible:=true;

  if
  (cdsayisi>=1)and(StrPos(pchar(form1.N1CDA1.caption),'(')
  =nil) then form1.N1CDA1.caption:=form1.N1CDA1.caption
  +' ('+cdler[0]+'.');
  if
  (cdsayisi>=2)and(StrPos(pchar(form1.N2CDA1.caption),'(')
  =nil) then form1.N2CDA1.caption:=form1.N2CDA1.caption
  +' ('+cdler[1]+'.');
  if
  (cdsayisi>=3)and(StrPos(pchar(form1.N2CDA2.caption),'(')
  =nil) then form1.N2CDA2.caption:=form1.N2CDA2.caption
  +' ('+cdler[2]+'.');
  if
  (cdsayisi>=4)and(StrPos(pchar(form1.N2CDA3.caption),'(')
  =nil) then form1.N2CDA3.caption:=form1.N2CDA3.caption
  +' ('+cdler[3]+'.');

  MyProcList := TStringList.Create;
  try
    GetProcessList(MyProcList);
    if MyProcList = nil then Exit;
    ara:=Form1.ProgramKapat1.Count;
    for i:=ara-1 downto 0 do
      Form1.ProgramKapat1.Delete(i);
    for i := 0 to MyProcList.Count - 1 do begin
      if (copy(MyProcList.Strings[i],2,1)=':') then
        begin

//ShowMessage(MyProcList.Strings[i]+'*'+form1.ProgramKapat
1.Items[0].Caption);
      itme:=TMenuItem.Create(Form1.ProgramKapat1);
      itme.Caption:=MyProcList.Strings[i];
      itme.OnClick:=form1.tikla;

Form1.ProgramKapat1.Insert(Form1.ProgramKapat1.Coun
t,itme);
      //itme.free;
    end;
  end;
  finally
    MyProcList.Free;
  end;

end;

procedure TForm1.N1CDA1Click(Sender: TObject);
begin
if ac1 then begin

```

```

mciSendString(pchar('open cdaudio!' +cdler[0]+' : alias
drive'+cdler[0]), nil, 0, 0);
mciSendString(pchar('set drive'+cdler[0]+' door open
wait'), nil, 0, 0);
ac1:=False;
end else
begin
mciSendString(pchar('set drive'+cdler[0]+' door closed
wait'), nil, 0, 0);
ac1:=true;
end;
end;

procedure TForm1.N2CDA1Click(Sender: TObject);
begin
if ac2 then begin
mciSendString(pchar('open cdaudio!' +cdler[1]+' : alias
drive'+cdler[1]), nil, 0, 0);
mciSendString(pchar('set drive'+cdler[1]+' door open
wait'), nil, 0, 0);
ac2:=False;
end else
begin
mciSendString(pchar('set drive'+cdler[1]+' door closed
wait'), nil, 0, 0);
ac2:=true;
end;
end;

procedure TForm1.N2CDA2Click(Sender: TObject);
begin
if ac3 then begin
mciSendString(pchar('open cdaudio!' +cdler[2]+' : alias
drive'+cdler[2]), nil, 0, 0);
mciSendString(pchar('set drive'+cdler[2]+' door open
wait'), nil, 0, 0);
ac3:=False;
end else
begin
mciSendString(pchar('set drive'+cdler[2]+' door closed
wait'), nil, 0, 0);
ac3:=true;
end;
end;

end;

procedure TForm1.N2CDA3Click(Sender: TObject);
begin
if ac4 then begin
mciSendString(pchar('open cdaudio!' +cdler[3]+' : alias
drive'+cdler[3]), nil, 0, 0);
mciSendString(pchar('set drive'+cdler[3]+' door open
wait'), nil, 0, 0);
ac4:=False;
end else
begin
mciSendString(pchar('set drive'+cdler[3]+' door closed
wait'), nil, 0, 0);
ac4:=true;
end;
end;
end;

function KillTask(ExeFileName: string): integer;
const
  PROCESS_TERMINATE=$0001;
var

```

```

ContinueLoop: BOOL;
FSnapshotHandle: THandle;
FProcessEntry32: TProcessEntry32;
begin
    result := 0;

    FSnapshotHandle := CreateToolhelp32Snapshot
        (TH32CS_SNAPPROCESS, 0);
    FProcessEntry32.dwSize := Sizeof(FProcessEntry32);
    ContinueLoop := Process32First(FSnapshotHandle,
        FProcessEntry32);

    while integer(ContinueLoop) <> 0 do
    begin
        if
            ((UpperCase(ExtractFileName(FProcessEntry32.szExeFile)
            ) =
                UpperCase(ExeFileName))
            or (UpperCase(FProcessEntry32.szExeFile) =
                UpperCase(ExeFileName))) then
            Result := Integer(TerminateProcess(OpenProcess(
                PROCESS_TERMINATE, BOOL(0),
                FProcessEntry32.th32ProcessID), 0));
            ContinueLoop := Process32Next(FSnapshotHandle,
                FProcessEntry32);
        end;

        CloseHandle(FSnapshotHandle);
    end;

    procedure TForm1.tikla(Sender: TObject);
    begin
        if MessageDlg(ExtractFileName((sender as
            TMenuItem).caption)+#13+#13+'Bu program kapatılsın
            mı?',
            mtConfirmation, [mbYes, mbNo], 0) = mrYes Then
            KillTask(ExtractFileName((sender as
            TMenuItem).caption));
        CDlistesi;
    end;

    procedure TForm1.FormCreate(Sender: TObject);
    var
        i: Integer;
    begin
        SetWindowLong(Application.Handle, GWL_EXSTYLE,
            WS_EX_TOOLWINDOW);
        CreateMutex(nil, FALSE, 'project1');
        if GetLastError = ERROR_ALREADY_EXISTS then Halt(0);

        nid.cbSize:=sizeof(nid);
        nid.Wnd:=Handle;
        nid.uID:=1;
        nid.uCallbackMessage:=wm_IconMessage;
        nid.hIcon:=application.Icon.Handle;
        nid.uFlags:=nif_Message or nif_Icon or nif_Tip;
        shell_notifyicon(NIM_ADD,@nid);
        strcpy(nid.sztip,pchar('CD Eject'));
        Shell_NotifyIcon(NIM_ADD,@nid);
        form1.Width:=0;
        form1.Height:=0;
        top:=0;
        left:=0;

        cdsayisi:=0;

```

```

CDlistesi;

end;

procedure TForm1.IconTray(var Msg:TMessage);
var pt:Tpoint;
begin
    if
        (msg.LParam=wm_rbuttondown)or(msg.LParam=wm_lbutt
        ondown) then
        begin
            getcursorpos(Pt);
            setforegroundwindow(Handle);
            PopUpMenu1.Popup(Pt.x,Pt.y);
        end;
    end;

    procedure TForm1.FormClick(Sender: TObject);
    var msg:Tmessage;
    begin
        icontray(msg);
    end;

    procedure TForm1.WindowsuTekrarA1Click(Sender:
        TObject);
    begin
        if MessageDlg('Windows TEKRAR açılsın
            mı?',mtConfirmation, [mbYes, mbNo], 0) = mrYes Then
            ExitWindows(EWX_REBOOT,0);
        end;

    procedure TForm1.WindowsuKapat1Click(Sender:
        TObject);
    begin
        if MessageDlg('Windows KAPANSIN mı?',mtConfirmation,
            [mbYes, mbNo], 0) = mrYes Then
            ExitWindowsEx(EWX_SHUTDOWN,0);
        end;

    procedure TForm1.Timer1Timer(Sender: TObject);
    begin
        ShowWindow (handle, sw_hide);
        timer1.Enabled:=false;
    end;

    procedure TForm1.Kapat1Click(Sender: TObject);
    begin
        Shell_NotifyIcon(NIM_DELETE,@nid);
        Application.Terminate;
    end;

    procedure TForm1.Belgelermensntemizle1Click(Sender:
        TObject);
    begin
        if MessageDlg('Belgeler TEMİZLENSİN
            mı?',mtConfirmation, [mbYes, mbNo], 0) = mrYes Then
            SHAddToRecentDocs(SHARD_PATH, nil);
        end;

    procedure TForm1.azele1Click(Sender: TObject);
    begin
        CDlistesi;
    end;

end.

```

Form üzerinde sadece bir Timer1 ve PopupMenu1 var. PopupMenu'yü oluşturmak için üzerine çift tıklayın ve sırayla Windows'u Tekrar Aç, Windows'u Kapat, 1.CD Aç, 2.CD Aç ve Kapat menülerini ekleyin. Bu fare sağ tuş menüsünün görev çubuğuna çıkacak olan simgeye sağ tıklayınca çıkmasını sağlayacağız.

Eksiklikler; CD'lerin adlarını alamama, çıkartılmış CD'yi hissedememe, başka ülke dillerini destekleme, dinamik kullanıcı tanımlı komutlar...

DOS için TSR, Win 3.1 için simge halinde çalışma ve Windows 9x için ise görev çubuğunda simge olarak devamlı çalışan hizmet programlar vardır. XP, 2003 ve üstünde de Hizmetler (Services) ile *.msc dosyaları istersek başlangıçta çalıştırabiliyor, istersek sonradan çalıştırabiliyoruz. .NET kullanarak kendi *.msc dosyalarımızı yapabiliriz. Artık NT tabanlı sistemler yaygın olmasına rağmen, çok az yazılım bu arabirimi tercih ediyor. En basitinden bir programın her zaman başlangıçta açılmasını isterseniz, "Başlat*Programlar*Başlangıç" menüsüne kısayolunu ekleyin.

Siz de bu menüye dinamik olarak eklenebilen silinebilen program kısayolları, yardımcı komutlar ekleyebilirsiniz. Dinamik deyince, akla bu değişen bilgileri nerede saklayacağımız geliyor. Registry'e, veritabanına, metin dosyasına veya *.ini – initiation (ilkeme) dosyalarına kayıt yapabiliriz. Bunların içinden bu tip basit programlar için en kullanışlı olanı *.ini'dir. Meraba.scr'de bir örneğini görebilirsiniz. Registry gereksiz yere şişmesinden dolayı kullanmaya gerek yoktur, ama Delphi'de kayıt ve okuma desteği doğal olarak vardır. "Ornekler\Delphi\regis\project1.dpr" örneğinde Windows kayıtlarının geniş kullanımını görebilirsiniz. Veritabanı kullanmak da basit işler için ağır ve BDE'den dolayı taşınabilirliği azdır. Metin dosyalarını okumak yerine *.ini kullanmak daha hoş, çünkü istediğiniz satıra rahatlıkla gidebilir, yeni veri rahatlıkla ekleyebilirsiniz. Delphi'nin ürettiği *.dsk ve *.opt dosyaları, msdos.sys, *.inf donanım bilgisi dosyaları da *.ini yapısındadır. XML - Extensible Markup Language (Genişletilebilir İşaretleme Dili) geleceğin internet ve yerel veritabanı olacak gözüküyor. HTML ile birlikte çalışarak veya Delphi içine Microsoft XML, v3.0 (Version 3.0) nesnesi eklenerek kullanabiliriz.

*.csv comma delimited – virgülle ayrılmış bilgileri Excel'de oluşturabilir okuyabilirsiniz. *.csv normal metin dosyası olduğu için, Delphi'de uzun ve benzer bilgilerle dolu olan verilerinizi bu tip dosyalara kaydedebilirsiniz. Tabii bilginin içeriğinde de ";" karakteri olmamalıdır; ama "" içine istediğinizi yazabilirsiniz. "a;b;a";333 gibi...

	A	B	C	D
1	s	23423	3	
2	dfsf	234	3	
3	sf	2342	3	
4	sfsdf	34234	3	
5	fs	234	3	
6	dfsf	234	3	
7	dfsf	234	3	
8	dfsd	234	3	
9	fs	234	3	
10	sdfsdf	234	3	
11				

Book1.csv - Not Defteri

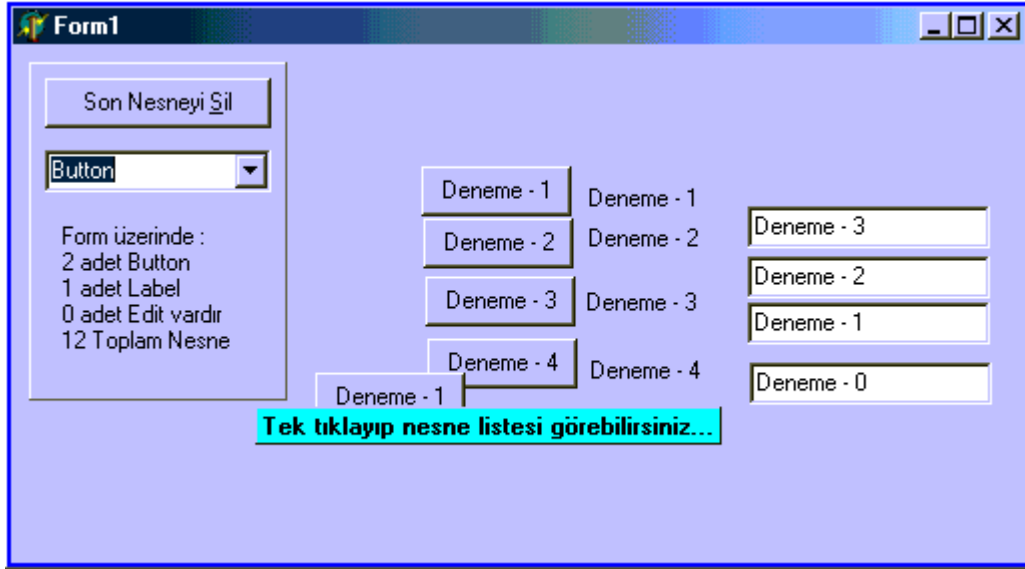
```
s;23423;3
dfsf;234;3
sf;2342;3
sfsdf;34234;3
fs;234;3
dfsf;234;3
dfsf;234;3
dfsd;234;3
fs;234;3
sdfsdf;234;3
```

Resim 5 - *.csv dosyalar

3. create (nesne oluşturma)

(Ornekler\Delphi\create)

Bu örnek nesne oluşturmak ile ilgili. Çalışma zamanında fare ile form üzerine tıklanınca düğme, metin kutusu ve etiket nesnesi oluşturuluyor. Tasarımda kendi oluşturduğumuz "Son Nesneyi Sil" düğmesi yani Button1 ile son nesneyi siliyoruz. Form üzerinde ayrıca PopupMenu1, Timer1, OpenFileDialog1 ve SaveFileDialog1 nesneleri vardır.



Resim 6 – Nesne oluşturmak ve silmek

```
object Form1: TForm1
  Left = 271
  Top = 104
  Width = 696
  Height = 480
  Caption = 'Form1'
  Color = clBtnFace
  OldCreateOrder = False
  PopupMenu = PopupMenu1
  OnCreate = FormCreate
  OnMouseDown = FormMouseDown
  PixelsPerInch = 96
  TextHeight = 13
  object Panel1: TPanel
    Left = 8
    Top = 8
    Width = 129
    Height = 169
    Cursor = crArrow
    Hint = 'Tek tıklayıp nesne listesi görebilirsiniz...'
    ParentShowHint = False
    ShowHint = True
    TabOrder = 0
    OnClick = Panel1Click
  object Label1: TLabel
    Left = 16
    Top = 80
    Width = 32
    Height = 13
    Caption = 'Label1'
  end
  object Button1: TButton
    Left = 8
```

```
    Top = 8
    Width = 113
    Height = 25
    Caption = 'Son Nesneyi &Sil'
    TabOrder = 0
    OnClick = Button1Click
  end
  object ComboBox1: TComboBox
    Left = 8
    Top = 44
    Width = 113
    Height = 21
    ItemHeight = 13
    TabOrder = 1
    Items.Strings = (
      '<Boş>'
      'Button'
      'Label'
      'Edit')
  end
end
object Timer1: TTimer
  Interval = 10
  OnTimer = Timer1Timer
  Left = 216
  Top = 16
end
object OpenFileDialog1: TOpenDialog
  DefaultExt = 'txt'
  Title = 'Form Aç'
  Left = 360
  Top = 16
end
```



```

object SaveDialog1: TSaveDialog
  DefaultExt = 'txt'
  Title = 'Form Kaydı'
  Left = 432
  Top = 16
end
object PopupMenu1: TPopupMenu
  Left = 504
  Top = 16
  object Yeni1: TMenuItem
    Caption = '&Yeni'
    OnClick = Yeni1Click
  end

```

Unit içine yazılacak kod:

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, ExtCtrls, Menus, Grids;

type
  TForm1 = class(TForm)
    Button1: TButton;
    ComboBox1: TComboBox;
    FormA1: TMenuItem;
    FormKaydet1: TMenuItem;
    Label1: TLabel;
    OpenFileDialog1: TOpenDialog;
    Panel1: TPanel;
    PopupMenu1: TPopupMenu;
    SaveDialog1: TSaveDialog;
    Timer1: TTimer;
    Yeni1: TMenuItem;
    procedure FormMouseDown(Sender: TObject; Button:
    TMouseButton; Shift: TShiftState; X, Y: Integer);
    procedure Button1Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormKaydet1Click(Sender: TObject);
    procedure FormA1Click(Sender: TObject);
    procedure Yeni1Click(Sender: TObject);
    procedure Panel1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations } //Kendi olaylarımız burada
    procedure tiklama(Sender: TObject);
    procedure tusbasimi(Sender: TObject; var Key: Char);
    procedure metin_cikisi(Sender: TObject);
    procedure metne_giris(Sender: TObject);
  end;
var
  Form1: TForm1;
  butt,labe,edit:integer;          //nesne sayıları
implementation
{$R *.DFM}

procedure TForm1.metin_cikisi(Sender: TObject);
begin  caption:='Form1'; end;

procedure TForm1.metne_giris (Sender: TObject);
var temp:TEdit;
begin
temp:=sender as TEdit;  caption:=temp.Text;
end;

```

```

end
object FormKaydet1: TMenuItem
  Caption = '&Form Kaydet'
  OnClick = FormKaydet1Click
end
object FormA1: TMenuItem
  Caption = 'Form &Aç'
  OnClick = FormA1Click
end
end
end
end

```

```

procedure TForm1.tusbasimi (Sender: TObject; var Key:
Char);
var temp:TEdit;
begin
temp:=sender as TEdit;  caption:=temp.Text+key;
//tuş okuyor ve aktarıyor
end;

```

```

procedure TForm1.tiklama (Sender: TObject);
begin
if sender is TButton then
  ShowMessage('Düğme Tıklandı!...');
if sender is TLabel then
  ShowMessage('Etiket Tıklandı!...');
end;

```

```

procedure TForm1.FormMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var  Dugmemiz: TButton;Etiket:TLabel;Metin:TEdit;
begin
if ssLeft in shift then begin
case combobox1.ItemIndex of
1: begin Dugmemiz := TButton.Create (self);
//yeni düğme oluşturuldu...
      Dugmemiz.Parent:= self;
      Dugmemiz.Caption:='Deneme - ' + IntToStr (butt);
      Dugmemiz.Top:= y;
      Dugmemiz.Left:= x;
      Dugmemiz.Name:=TButton.ClassName+IntToStr
(butt);
      Dugmemiz.OnClick:=tiklama;
      end;
2: begin Etiket := TLabel.Create (self);
//yeni etiket oluşturuldu...
      Etiket.Parent:= self;
      Etiket.Caption:='Deneme - ' + IntToStr (labe);
      Etiket.Top:= y;
      Etiket.Left:= x;
      Etiket.Name:=TLabel.ClassName+IntToStr (labe);
      etiket.OnClick:=tiklama;
      end;
3: begin Metin := TEdit.Create (self);
//yeni metin kutusu oluşturuldu...
      Metin.Parent:= self;
      Metin.text:='Deneme - ' + IntToStr (edit);
      Metin.Top:= y;
      Metin.Left:= x;
      Metin.Name:=TEdit.ClassName+IntToStr (edit);
      Metin.OnKeyPress:=tusbasimi;
      Metin.OnEnter:= metne_giris;

```

```

    Metin.OnExit:=metin_cikisi;
end;
end;
end;
end;

procedure TForm1.Button1Click(Sender: TObject);
//Son nesneyi siler
begin
    if Form1.ComponentCount > 11 then
    //temel nesneler dışındakiler silinebilir
    begin
        Components[Form1.ComponentCount-1].Free;
    end;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
var i:integer;Temp: TComponent;
begin
    butt:=0;labe:=0;edit:=0;
    for i:=0 to Form1.ComponentCount-1 do begin
        Temp:=Components[i];
        if Temp is TButton then inc(butt);
        if Temp is TLabel then inc(labe);
        if Temp is TEdit then inc(edit);
    end;
    label1.Caption:='Form üzerinde :'+#13+inttostr(butt)+' adet
    Button'+#13+inttostr(labe)+' adet
    Label'+#13+inttostr(edit)+' adet Edit vardır'
    +#13+inttostr(ComponentCount)+' Toplam Nesne';
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    ComboBox1.ItemIndex:=0;
    Application.HintColor:=claquea;
    Application.HintHidePause:=3000; //hint bekleme zamanı
    Application.HintPause:=300; //hint çıkma zamanı
end;

procedure TForm1.FormKaydet1Click(Sender: TObject);
var dosya :TFileStream;i:integer;
begin
    if SaveDialog1.Execute then
    begin
        // WriteComponentResFile(SaveDialog1.filename,Self); /*.dfm
        olarak da kayıt yapmak mümkün

    dosya:=TFileStream.Create(SaveDialog1.FileName,fmOpe
    nWrite or fmCreate);
        for i:=11 to form1.ComponentCount-1 do
        begin
            dosya.WriteComponent(Components[i]);

```

```

        end;
        dosya.Free;
    end;
end;

procedure TForm1.Yeni1Click(Sender: TObject);
var i,nesnesayisi:integer;
begin
    if form1.ComponentCount>11 then
    //temel nesneler dışındakiler silinebilir
    begin
        nesnesayisi:=form1.ComponentCount-1;
        for i:=nesnesayisi downto 11 do
        //ters yön olmazsa silinemez!...
            Components[i].Free;
        end;
    end;
end;

procedure TForm1.FormA1Click(Sender: TObject);
var dosya :TFileStream;nesne:TComponent;
    dugmemiz:TButton; etiket:TLabel; metin:TEdit;
begin
    if OpenFileDialog1.Execute then
    begin
        Yeni1Click(self);

    dosya:=TFileStream.Create(OpenDialog1.FileName,fmOpe
    nRead);
        while dosya.Position<dosya.size do
        begin
            nesne:=dosya.ReadComponent(nil);
            InsertControl(nesne as Tcontrol);
            //sadece görünüm olarak varoluyor...?!
        end;
        dosya.Free;
    end;
end;

procedure TForm1.Panel1Click(Sender: TObject);
var i:integer;nesneler:string;
begin
    for i:=0 to form1.ComponentCount-1 do
        nesneler:=nesneler+inttostr(i+1)+#9+Components[i].Name
        + ' / '+Components[i].className+#13;
    ShowMessage(nesneler);
end;

initialization
//formdaki tüm nesne tipleri dosyalama için hazırlanıyor
    RegisterClasses ([TPanel, TCombobox, TLabel, TEdit,
    TButton]);
end.

```

“Ornekler\Delphi\surukle” ve bu örnek için eklemek gerekirse Label veya benzeri nesneleri kullanıcı programı çalıştırırken oluşturup silebilir. Nesne özellikleri *.txt veya *.dfm olarak bir dosyaya kaydedilerek değişik tiplerde etiket dökümü alınabilir. Başlık, koordinat, yükseklik, genişlik gibi belli başlı özelliklerini değiştirip kaydetmek yeterlidir. Düğmeleri için ise ayrıca *.txt veya *.pas olarak kod kısımları kaydedilebilir. Şimdilik tekrar açtığımız form öğeleri kod kısımlarını hissedemiyor, sadece görünüm olarak var oluyorlar. Belki kendi dosya kaydetme yöntemimiz ile yapsaydık daha rahat kontrolleri idare edebilirdik. Örnekle aynı klasörde Yunus isimli bir projem var. Burada birçok eksiği giderdim. Hatta minik bir Türkçe Delphi yaptık da diyebiliriz.

Aslında yavaş yavaş kendi Delphi component – nesnenizi oluşturmaya da yaklaşıyoruz. Fakat nesne tasarımı ile hiç ilgilenmedim, burada örnek veremiyorum.

4. Dergilerim (basit veritabanı uygulaması)

(Ornekler\Delphi\Dergilerim\dergilerim.dpr)

Amaç, bilgisayar dergilerinde bulunan belli konulardaki yazıları listelemek ve aramak. Veritabanı uygulamamız.

Bunun için Delphi'nin veritabanı yöneticisi olan BDE – Borland Database Engine Administrator programını çalıştırıp bir takma ad belirliyoruz. "dergiler" adında, path – yol tanımı projemizin olduğu dizin olan bir alias – takma ad oluşturuyoruz. Alias tanımlamadan da veritabanı ile uğraşabiliriz. Fakat daha sonra proje bitince setup - kurulum hazırlarken bize gerekli olması. Zaten tasarım anında da bize faydası, yeni bir table – tablo eklediğimizde "Database=dergiler" deyince "TableName" içinde bir açılır kutu içine o dizindeki tüm *.db ve *.dbf dosyaları getirmesi. Böylece "C:\Belgelerim\Delphi\Örnek\deneme.db" gibi sabit bilgilere dayalı işlem yapmıyoruz.

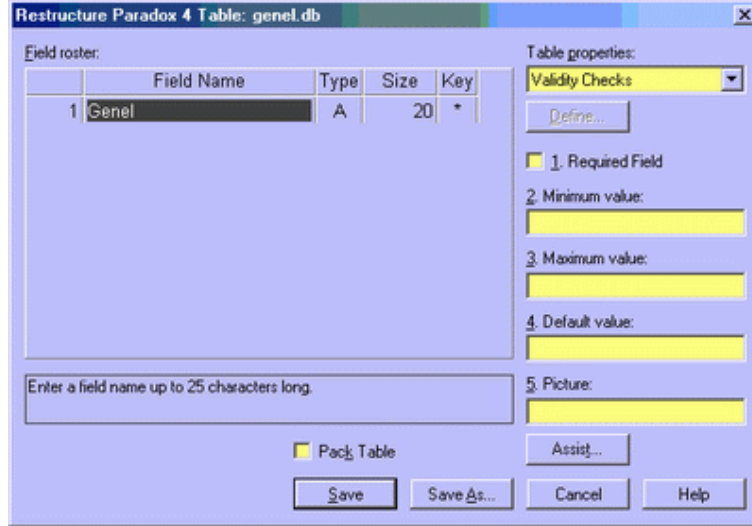
Üç veritabanı dosyamız var. genel.db (paradox 4), adlari.db (paradox 4) ve veri.dbf (dbase IV tipinde). Aslında böyle farklı farklı tipte hazırlamak mecburiyetinde değilsiniz. Tek veritabanı da iş görür, fakat yeni dergi adı ekleme, silme için adlari.db isminde veritabanı, genel kısmı için de genel.db oluşturdum.

Dergi adlarının hiç değişmeyeceğini düşünseydik, DBGrid'in istediğimiz sütunundaki bilgiye PickList özelliği sayesinde sabit bilgiler girebilirdik. DBGrid'e çift tıklayın, çıkan menüye sağ tıklayıp "Add All Fields" deyin. Eklenen alan adlarından istediğinizi seçip PickList'ini doldurun.

Yapıları oluşturduktan sonra veri.dbf içinde bir işlemiz daha kalıyor. Index – sıralama oluşturmak. Alan adlarından Genel, Yıl ve DergiAdi alanlarına bağlı sıralama tiplerimiz var. Yıl bu arada önce yıla sonra da aya göre karma indeks oluşturmamızı sağlayacak.

Paradox'da Key sütunundaki "*" belirtici ile primary index – birincil indeks, "Table Properties*Secondary Index" ile de ikincil indeks oluşturuyoruz. Key – anahtar kısmının bir amacı da aynı kayıttan bir daha yapılamamasını sağlamaktır. Yani iki tane Ahmet ismi kaydedilemez. Secondary Index 'de böyle bir kısıtlama yoktur.

Programımızı hazırlarken de indeksler arasında geçiş yapabilmek için Combobox – açılır kutu kullanıyoruz. Combo combine – birleştir kelimesinden türemiştir. Aramak için de tablo filtreleme yapıyoruz. Aramak için wind* yazıp, Enter tuşuna basın, wind ile başlayan konuları filtrelediğini gördünüz. Maalesef *98* gibi yazıp içinde 98 geçen konuları arayamıyoruz. Bu şekilde aramak isterseniz Query – sorgu nesnesi kullanmanız gerekiyor. SQL* içindeki **Like** komutu tam bize göre.



* Ek 2 – SQL kısmına bakınız.

Restructure Paradox 4 Table: adlari.DB

Field roster:

	Field Name	Type	Size	Key
1	Adi	A	20	*

Enter a field name up to 25 characters long.

☐ Pack Table

Table properties:

Validity Checks

☐ 1. Required Field

2. Minimum value:

3. Maximum value:

4. Default value:

5. Picture:

Restructure dBASE IV Table: veri.DBF

Field roster:

	Field Name	Type	Size	Dec
1	DERGIADI	C	20	
2	YIL	N	4	0
3	AY	N	2	0
4	GENEL	C	20	
5	KONU	C	30	
6	SAYFA	N	3	0

Enter field name of 10 characters or less, beginning with a letter. Use only A-Z, 0-9, or _.

☐ Record lock

☐ Info size

☐ Pack Table

Table properties:

Indexes

DERGIADI
GENEL
YIL

Resim 7 - Veritabanlarının yapıları

Dergi İndeksleme

Veri Girişi Konu Filtreleme Diğer Girişler

Konu Arama=

Yıl=	Ay=	Dergi Adı=	Genel=	Konu=	Sayfa=

Dergi İndeksleme

Veri Girişi Konu Filtreleme Diğer Girişler

Dergi Adı Genel Konu Adı

Bilgisayar Pazarı Diger

Yıl	Ay	Dergi Adı	Genel	Konu	Sayfa
1998	3	Byte	İşletim Sistemi	NT 4.0	54
1998	3	Byte	Uygulama	FDisk	38
1998	3	Byte	Uygulama	Excel Denklem Çözümü	109
1998	3	Byte	Programlama	Delphi - Component	118
1998	3	Byte	Oyun	Blood II & Diablo II	142
1996	9	Chip	Donanim	PCI-Chipset	72
1996	9	Chip	Donanim	MPEG ve PC video	90
1996	9	Chip	Oyun	Earth Ziege 2	178
1996	9	Chip	Donanim	AGP	94
1996	12	Chip	Donanim	RAM hakkında	68
1996	12	Chip	Donanim	PC hakkında	138
1996	12	Chip	İşletim Sistemi	NT 4.0	148
1997	1	Chip	Donanim	BIOS	156
1997	1	Chip	Donanim	Aygıt Sürücüler Kurulumu	47
1997	1	Chip	Donanim	InkJet Prensipleri	80
1997	2	Chip	Donanim	JoyStick	106
1997	3	Chip	Donanim	Yedekleme Üniteleri	102
1997	3	Chip	Donanim	100 Sorusu	60
1997	3	Chip	Uygulama	Virus Programları	50
1997	3	Chip	Diğer	PC ve Sağlık	34
1997	3	Chip	Donanim	Modem Prensipleri	138

Resim 8 – Dergilerim tasarım görünümü

Hala işimiz bitmedi. Çünkü DergiAdı ve Genel sütunundaki bilgileri Combobox listesi şeklinde veri girişi yapmamız daha iyi. Zaten bu sebeple ayrı iki veritabanı da oluşturduk. Dergi adlarını adlari.db dosyasından, genel sütununun verilerini de genel.db dosyasından okuyacak.

Table1 bizim ana verilerin tutulduğu yer. Üzerine çift tıklayıp "Add all fields" deyin. Ama buradaki DergiAdı ve Genel bizi ilgilendirmiyor. Çünkü kendi alanımızı oluşturacağız. Yine aynı yere sağ tıklayıp "New field..." deyin. Amacımız Lookup field – arama alanı oluşturmak. İlk alanımız "Ad" isminde, diğeri "Geneli".

New Field

Field properties:

Name: Component:

Type: Size:

Field type:

☐ Data ☐ Calculated ☒ **Lookup**

Lookup definition:

Key Fields: Dataset:

Lookup Keys: Result Field:

OK Cancel Help

Resim 9 – Yeni alanlar tanımlamak – Lookup Fields

unit dergiform;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs,
Db, DBTables, ExtCtrls, DBCtrls, Grids, DBGrids,
StdCtrls, Buttons,
ComCtrls, Mask, ToolWin;

type

```
TForm1 = class(TForm)
  ComboBox1: TComboBox;
  DataSource1: TDataSource;
  DataSource2: TDataSource;
  DataSource3: TDataSource;
  DataSource4: TDataSource;
  DBEdit1: TDBEdit;
  DBEdit2: TDBEdit;
  DBGrid1: TDBGrid;
  DBGrid2: TDBGrid;
  DBNavigator1: TDBNavigator;
  DBNavigator2: TDBNavigator;
  DBNavigator3: TDBNavigator;
  Edit1: TEdit;
  Label1: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  PageControl1: TPageControl;
  SpeedButton1: TSpeedButton;
  SpeedButton2: TSpeedButton;
  SpeedButton3: TSpeedButton;
  Table1: TTable;
  Table1Ad: TStringField;
  Table1AY: TSmallintField;
  Table1DERGIADI: TStringField;
  Table1GENEL: TStringField;
  Table1Geneli: TStringField;
  Table1KONU: TStringField;
  Table1SAYFA: TSmallintField;
  Table1YIL: TSmallintField;
```

```
Table2: TTable;
Table2AY: TSmallintField;
Table2DERGIADI: TStringField;
Table2GENEL: TStringField;
Table2KONU: TStringField;
Table2SAYFA: TSmallintField;
Table2YIL: TSmallintField;
Table3: TTable;
Table3Adi: TStringField;
Table4: TTable;
TabSheet1: TTabSheet;
TabSheet2: TTabSheet;
TabSheet3: TTabSheet;
ToolBar1: TToolBar;
ToolBar2: TToolBar;
procedure FormCreate(Sender: TObject);
procedure ComboBox1Change(Sender: TObject);
procedure Edit1KeyPress(Sender: TObject; var Key:
Char);
procedure SpeedButton1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

Form1: TForm1;

implementation

{ \$R *.DFM }

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  table1.open;
  combobox1.ItemIndex:=0;
end;
```

```
procedure TForm1.ComboBox1Change(Sender: TObject);
begin
  case combobox1.ItemIndex of
    0: table1.IndexName:="";
```

```
1:table1.IndexName:='Yil';
2:table1.IndexName:='DergiAdi';
3:table1.IndexName:='Genel';
end;
end;

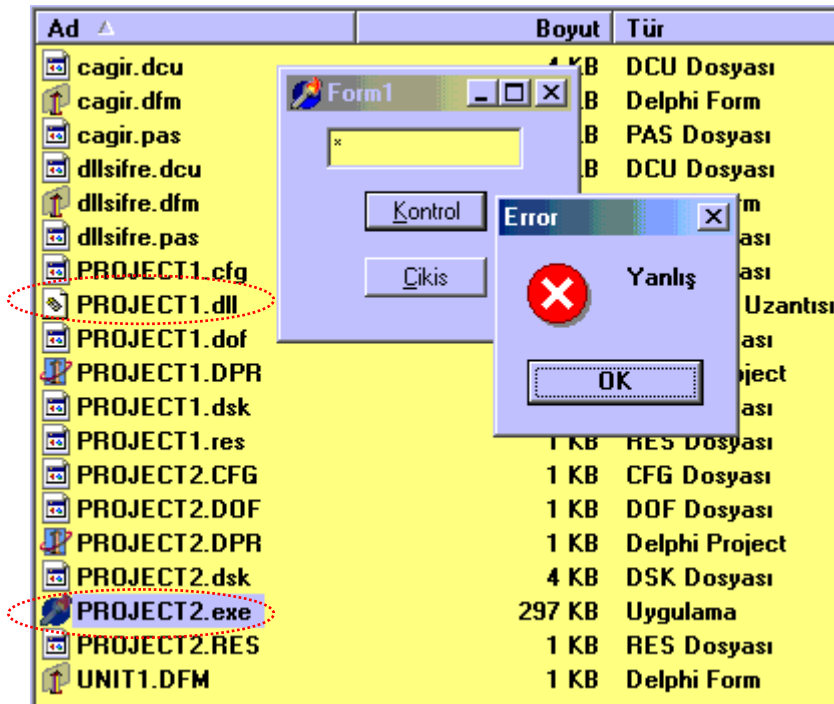
procedure TForm1.Edit1KeyPress(Sender: TObject; var
Key: Char);
begin
if key=#13 then begin
table2.Filter:='konu="'+edit1.Text+'";
```

```
table2.Filteroptions:=[focaseinsensitive];
table2.Filtered:=true;
end;
end;

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
form1.Close;
end;
end.
```

5. DLL (Dynamic Linking Library)

(Ornekler\Delphi\DLL\Project2.dpr)



Resim 10 – DLL ile şifre kontrolü

Dynamic Linking Library – dinamik bağ kütüphanesi genellikle projemizin yardımcı fonksiyonlarının tutulduğu yer olarak düşünebiliriz. C'deki *.h (header - başlık), Pascal'daki *.tpu (unit) gibi. Başka bir programlama dili ile de *.dll dosyanızı kullanabilirsiniz. Delphi dilinde farkında olmadan birçok Windows fonksiyonunu çağırıyoruz. *.dll çağırmak ile Win32API çağırmak bu sebeple aynı anlamda da kullanılabilir. Fakat derleyiciye çağıracağınız API fonksiyonunun tüm parametreleri ve tipleri hakkında tam bilgi vermek gereklidir. Yanlış tanımlanmış bir tip Windows'umuzda bir GPF – General Protection Failure penceresi çıkmasına sebep olur.

DLL'ler hafızadan tasarruf amacıyla kullanılabilir. Çünkü bir kere yüklenen bir *.dll dosyayı birden fazla program kullanabilir. Programınızı tekrar derlemek yerine sadece yeni bir *.dll sürümü oluşturmak yeterlidir. Yani patch – yamalar demek ki bu şekilde yapılabiliyormuş. İkon, metin veya bitmap deposu olarak da kullanılabilir. "Windows\System" dizinindeki pifmgr.dll, moricons.dll, iconlib.dll ve shell32.dll içinde birçok simge görmüşsünüzdür.

Eğer Windows programcılığı ile ilgili geniş bir kaynak arıyorsanız, tek yapmanız gereken Delphi Yardım kısa yollarındaki MS SDK Help Files klasörünü incelemektir.

VB dilinde bir Win32API çağırmak için modül içine şu şekilde yazabilirsiniz:

```
Declare Function GetDiskFreeSpace Lib "kernel32.dll" Alias "GetDiskFreeSpaceA" _
  (ByVal lpRootPathName As String, _
  ByVal lpSectorsPerCluster As Long, _
  ByVal lpBytesPerSector As Long, _
  ByVal lpNumberOfFreeClusters As Long, _
  ByVal lpTotalNumberOfClusters As Long _
  ) As Long
```

Windows kernel – çekirdek (kernel32.dll), user – kullanıcı (user32.dll) ve GDI – Graphical Device Interface (gdi32.dll) merkezli olarak çalışır. Diğer kütüphaneler ise denetimler, donanım sürücüsü, yazı tipleri gibi uzantılardır. Tüm çalışan programlar bu *.dll dosyalara bağlıdır.

Delphi'de Windows kütüphanesinde şuna benzer API'ler tanımlanmıştır:

```
{$EXTERNALSYM Pie}
```

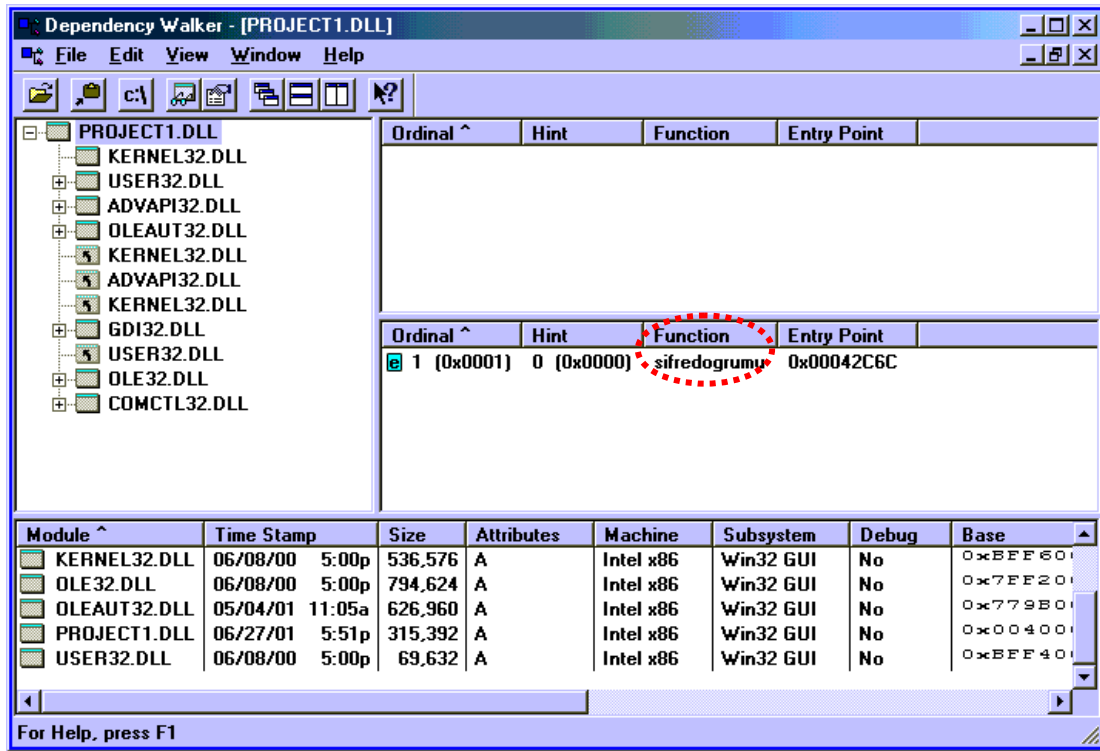


```

function Pie(DC: HDC; X1, Y1, X2, Y2, X3, Y3, X4, Y4: Integer): BOOL; stdcall;
...
const
...
kernel32 = 'kernel32.dll';
{$EXTERNALSYM version}
version = 'version.dll';
comctl32 = 'comctl32.dll';
gdi32 = 'gdi32.dll';
opengl32 = 'opengl32.dll';
user32 = 'user32.dll';
...
implementation
...
function CopyFile; external kernel32 name 'CopyFileA';
function CopyFileEx; external kernel32 name 'CopyFileExA';
function CreateConsoleScreenBuffer; external kernel32 name 'CreateConsoleScreenBuffer';
function CreateDirectoryA; external kernel32 name 'CreateDirectoryA';

```

Üstteki Windows unit'ine dikkat ederseniz tüm Windows fonksiyonlarının elimizin altında olduğunu görebilirsiniz. Bir DLL'de hangi fonksiyon veya prosedürler var öğrenmek için Dependency Walker (Diger\Extras\DEPENDS.EXE) adlı programını kullanmanızı tavsiye ederim:

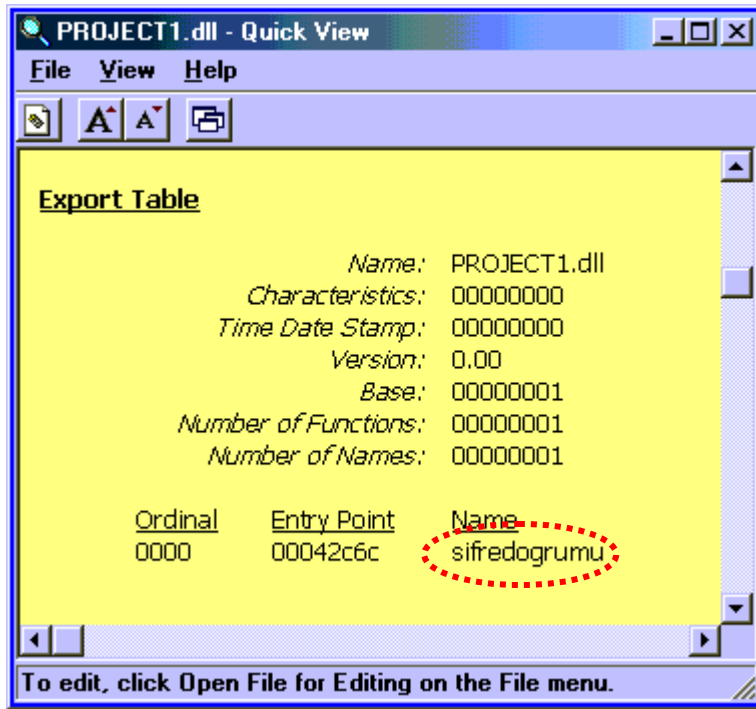


Resim 11A – Dependency Walker (Bağımlılık gezgini)

"rundll32.exe" DLL dosyaları çalıştırabiliriz. Rundll32 için aşağıdaki gibi, önce *.dll adı ardından virgülle fonksiyon adı olacak şekilde bir kısayol oluşturabilirsiniz ve "Başlat*Çalıştır..." ile deneyebiliriz:

rundll32.exe shell32.dll, Control_RunDLL directx.cpl

Windows'ta QuickView ile inceleyebiliriz. Proje1.dll dosyamızın Export Table bölümü:



Resim 11B – Export Table kısmı

İki projemiz var. Önce birincisini derleyin (Ctrl+F9). Project1.dll dosyasını oluşturdunuz. Diğer projede ise, sadece bu *.dll dosyamıza şifreyi yollayan metin kutusu var.

"File*New...*DLL" komutu ile Project1.dpr ve "File*New...*Form" diyerek de dllsifre.pas dosyamızı oluşturuyoruz :

```
library Project1;
uses
  dllsifre in 'dllsifre.pas' {kontrolform};
exports
  sifredogrumu;
{$R *.RES}
begin
end.
```

dllsifre.pas dosyamızın kodları:

```
unit dllsifre;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  StdCtrls;

type
  TForm1 = class(TForm)

  private
    { Private declarations }
  public
    { Public declarations }
    sifre:string;
  end;

var
  kontrolForm: TForm;
```

```
function sifredogrumu(sifreyial:string):boolean;export;
implementation
```

```
{$R *.DFM}
```

```
function sifredogrumu(sifreyial:string):boolean;
begin
  result:=false;
  kontrolform:=tform.create(application);
  //aslinda üstteki satır gereksiz, ama form create hoşunuza
  gidebilir...
  try
    if (sifreyial='B') then
      begin
        // kontrolform.showmodal;
        messagedlg('Doğru',mtInformation,[mbOk],0);
        result:=true;
      end
    else
      messagedlg('Yanlış',mtError,[mbOk],0);
  finally
    kontrolform.free;
  end;
end;
end.
```

Diğer projemiz olan Project2.dpr ise normal uygulama olarak oluşturuyoruz:

```
program Project2;
uses
```

```

Forms,
cagir in 'cagir.pas' {Form1};
{$R *.RES}
begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.

```

cagir.pas dosyamızın kodları:

```

unit cagir;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  StdCtrls;

type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Button1: TButton;
    Button2: TButton;
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public

```

```

    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.Button2Click(Sender: TObject);
begin
  close;
end;

procedure TForm1.Button1Click(Sender: TObject);
var tutacak:thandle;
    sifresonuc:function(sifreyial:string):boolean;
begin
  try
    tutacak:=loadlibrary('project1.dll');
    @sifresonuc:=getProcAddress(tutacak,'sifredogrumu');
    sifresonuc(edit1.text);
  finally
    freelibrary(tutacak);
  end;
end;

end.

```

6. Meraba (resimli ekran koruyucusu)

(Ornekler\Delphi\Meraba\Project1.dpr)

Windows Millennium içinde gelen "Resimlerim Ekran Koruyucusu" 'na benzeyen bir ekran koruyucusu. Bizim ekran koruyucumuz *.bmp ve *.jpg dosya içeren bir klasördeki resimleri rasgele veya sıralı gösteriyor.

Windows'taki ekran koruyucusu şifresini alamıyor. Şifre "meraba.ini" dosyasının içinde aynen görülebilir, güvenlik gelişmemiş. Program sonunda ek bir program olarak basit bir şifreleme algoritması veriyorum. Diğer resim formatları, müzik çalması, resimlerin geçiş efektlerini size bırakıyorum. Hatta daha ileri gidip alt klasörlerin de resimlerini dahil etmek isteyebilirsiniz, makrolardaki dosyalistesi.xls ve Delphi'deki Dosyalistesi örneklerinden faydalanabilirsiniz.

Temel olarak bütün ekran koruyucuları normal *.exe dosyalarıdır. Ayarları görmek için "/c", ön izleme yapmak için "/s" parametresi isterler. "Windows" veya "Windows\System" klasörüne kurulurlar. Ekran koruyucu dosyalarına sağ tıklandıklarında Sına, Yapılandır ve Yükle gibi seçenekler çıkar.

Derlenince oluşan Project1.exe'yi ekrankoruyucum.scr gibi tekrar adlandırın. Burada yine bir hata oluşuyor. Üzerine çift tıklanınca meraba.ini eğer aynı klasörde yok ise hata mesajları ile ekranınız dolabilir. En iyisi "Windows" içine *.scr dosyamızı atıp, "Bilgisayarım*Denetim Masası*Görüntü*Ekran Koruyucusu" 'ndan *.scr'mizi seçip, Ayarlar ile resim klasörümüzü belirtmek. Böylece meraba.ini oluşuyor ve sorun çıkmadan Önizleme yapılabilir. Aslında meraba.ini'yi silip de Önizleme komutunu verseniz kendi hata mesajımız çıkıyor ve ekran koruyucu kendini kapatıyor.

Form1'de resimler gösteriliyor, Form2 ayarlar, Form3 şifre ve AboutBox hakkında için kullanılıyor.

Proje dosyamız:

program Project1;

uses

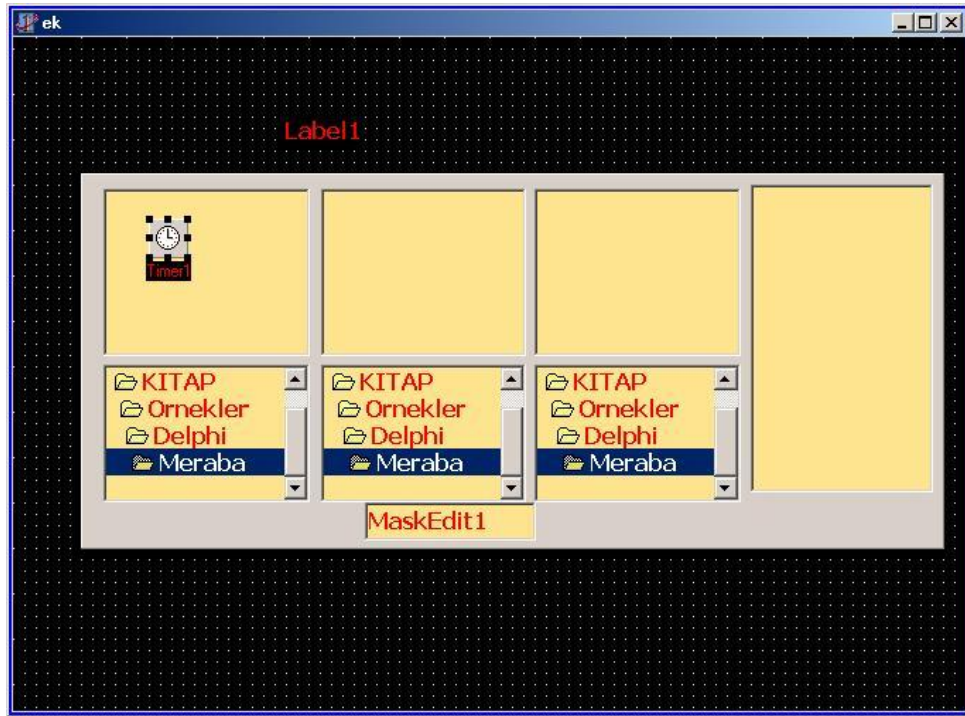
Forms,
Unit1 in 'Unit1.pas' {Form1},
Unit2 in 'Unit2.pas' {Form2},
About in 'about.pas' {AboutBox},
Unit3 in 'Unit3.pas' {Form3};

{ \$R *.RES }

var a:string;
begin

Application.Initialize;
Application.Title := 'Meraba 1.0';
a:= paramstr(1);
if (a[2]='c') or (a[2]='C') then //eğer ayar yapılmak isteniyorsa sadece 2 form create edilecek
begin
Application.CreateForm(TForm2, Form2);
Application.CreateForm(TAboutBox, AboutBox);
Application.Run;
end
else if (a='/s') or (a='/S') then //sadece çalıştırılacak ise ana form ve şifre kutusu create edilecek
begin
Application.CreateForm(TForm1, Form1);
Application.CreateForm(TForm3, Form3);
Application.Run;
end;
end.

Görüldüğü gibi program çalışmadan önce parametreye bağlı olarak ya Form1, ya da Form2 ekrana geliyor.



Form1'in kodları:

```

unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, DIB, inifiles, StdCtrls, FileCtrl, jpeg, Mask;

type
  TForm1 = class(TForm)
    DirectoryListBox1: TDirectoryListBox;
    DirectoryListBox2: TDirectoryListBox;
    DirectoryListBox3: TDirectoryListBox;
    FileListBox1: TFileListBox;
    FileListBox2: TFileListBox;
    FileListBox3: TFileListBox;
    Image1: TImage;
    Label1: TLabel;
    ListBox1: TListBox;
    MaskEdit1: TMaskEdit;
    Panel1: TPanel;
    Timer1: TTimer;
    procedure FormKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
    procedure Image1MouseDown(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
    procedure FormShow(Sender: TObject);
    procedure AyarOku;
    procedure Timer1Timer(Sender: TObject);
    procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
  private
    { Private declarations }
    htb:HWND;
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  duration,sayac,sayac_ic:integer;

```

```
directry:string;rndm,stret,pw:boolean;
fname,pws:string;mh:THandle;
oldval:longint;
```

implementation

```
uses Unit3; //şifre kutusu
```

```
{$R *.DFM}
```

```
procedure TForm1.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
```

```
begin
  if pw then
    form3.show
  else
    begin
      ShowWindow(htb,SW_SHOW);
      SystemParametersInfo(97,word(false),@oldval,0);
      close;
    end;
end;
```

```
procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
```

```
begin
  if pw then
    form3.show
  else
    begin
      ShowWindow(htb,SW_SHOW);
      SystemParametersInfo(97,word(false),@oldval,0);
      close;
    end;
end;
```

```
procedure TForm1.AyarOku;
```

```
var dizin:string;
    ayarini:TIniFile;
begin
  dizin:=extractfilepath(application.ExeName)+'\Meraba.ini';
  ayarini:=tinifile.create(dizin);
  duration:=ayarini.readinteger('Meraba1.0','Duration',-99);
  directry:=ayarini.ReadString('Meraba1.0','Directory','');
  rndm:=ayarini.ReadBool('Meraba1.0','Random',True);
  stret:=ayarini.ReadBool('Meraba1.0','Stretch',True);
  pw:=ayarini.ReadBool('Meraba1.0','PW',True);
  pws:=ayarini.ReadString('Meraba1.0','PWs','');
  ayarini.Free;
end;
```

```
procedure TForm1.FormShow(Sender: TObject);
```

```
var i:integer;sayac1,sayac2,sayac3:integer;
    dizin:string;
begin
  dizin:=extractfilepath(application.ExeName)+'\Meraba.ini';
  if FileExists(dizin) then begin
    ShowWindow(application.Handle,SW_HIDE);
    htb:=FindWindow('Shell_Traywnd',nil);
    ShowWindow(htb,SW_HIDE);
    SystemParametersInfo(97,word(true),@oldval,0);
```

```
form1.Cursor:=crNone;
ayaroku;
randomize;
```

```

timer1.Interval:=duration*1000;
timer1.Enabled:=true;
DirectoryListBox1.Directory:=directry;
DirectoryListBox2.Directory:=directry;
DirectoryListBox3.Directory:=directry;
MaskEdit1.Text:=pws;
image1.Stretch:=stret;
sayac:=0;
sayac1:=0;
sayac2:=0;
sayac3:=0;
FileListBox1.Mask:= '*.bmp';
FileListBox1.Update;
FileListBox2.Mask:= '*.jpg';
FileListBox2.Update;
FileListBox3.Mask:= '*.jpeg';
FileListBox3.Update;
sayac1:=FileListBox1.Items.Count; //bmp
sayac2:=FileListBox2.Items.Count; //jpg
sayac3:=FileListBox3.Items.Count; //jpeg
listbox1.Clear;
if sayac1 <>0 then
  for i:=0 to sayac1-1 do
    ListBox1.Items.Add(FileListBox1.Items.Strings[i]);
  if sayac2 <>0 then
    for i:=0 to sayac2-1 do
      ListBox1.Items.Add(FileListBox2.Items.Strings[i]);
    if sayac3 <>0 then
      for i:=0 to sayac3-1 do
        ListBox1.Items.Add(FileListBox3.Items.Strings[i]);
      sayac:=sayac1+sayac2+sayac3;
      sayac_ic:=1;
      if sayac=0 then
        begin
          timer1.Enabled:=false;
          label1.Visible:=true;
          label1.Caption:='Please choose a directory which has photos!'
        end else begin
          image1.Picture.LoadFromFile(listbox1.Items.Strings[0]);
          image1.Update;
        end;
      end
    else begin
      timer1.Enabled:=false;
      ShowWindow(htb,SW_SHOW);
      SystemParametersInfo(97,word(false),@oldval,0);
      if Application.MessageBox(
        'You must choose your picture directory first'+#13+
        'from Control Panel*Display*Screen Saver*Settings' ,
        'Error', MB_OK) = IDOK then
        begin
          close;
        end;
      end;
    end;
  end;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
var fname:string;rastgele:integer;
begin
if rndm then
  begin
    rastgele:=random(listbox1.Items.Count);
    fname:=directry+'\'+listbox1.Items.Strings[rastgele];
    image1.Picture.LoadFromFile(fname);
    image1.Update;
  end
end

```

```

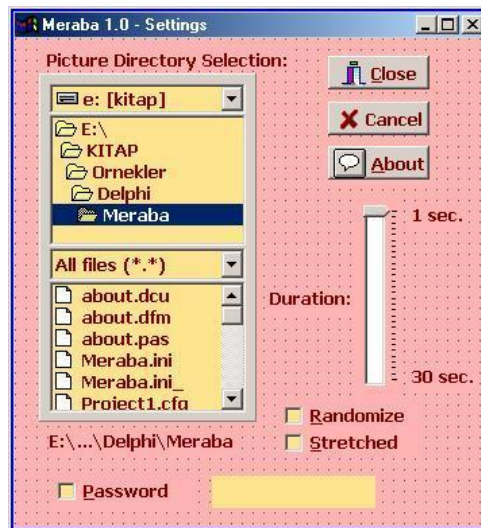
else
begin
if sayac_ic=listbox1.Items.Count then sayac_ic:=0;
fname:=directry+'\' +listbox1.Items.Strings[sayac_ic];
image1.Picture.LoadFromFile(fname);
image1.Update;
inc(sayac_ic);
end;
end;

procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
if pw then
canclose:=false
else
canclose:=true;
end;

//program windows'da 1 kere çalışabilir
initialization
mh:=createmutex(nil,False,pchar(Application.title));
if getlasterror=ERROR_ALREADY_EXISTS then halt;
finalization
if mh<>0 then CloseHandle(mh);

end.

```



Form2'nin kodları

```

unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, FileCtrl, ExtCtrls, ComCtrls, IniFiles, Mask;

type
  TForm2 = class(TForm)
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    CheckBox3: TCheckBox;
    DirectoryListBox1: TDirectoryListBox;
    DriveComboBox1: TDriveComboBox;

```



```

FileListBox1: TFileListBox;
FilterComboBox1: TFilterComboBox;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
MaskEdit1: TMaskEdit;
Panel1: TPanel;
TrackBar1: TTrackBar;
procedure BitBtn1Click(Sender: TObject);
procedure AyarYaz;
procedure AyarOku;
procedure BitBtn2Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure FormKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);
procedure CheckBox3Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form2: TForm2;

implementation

uses About;

{$R *.DFM}

procedure TForm2.BitBtn1Click(Sender: TObject);
begin
  AyarYaz;
close;
end;

procedure TForm2.AyarOku;
var dizin:string;
    duration:integer;directry,pws:string;
    rndm,stret,pw:boolean;
    ayarini:TIniFile;
begin
  dizin:=extractfilepath(application.ExeName);
  ayarini:=tinifile.create(dizin+'\Meraba.ini');
  duration:=ayarini.readinteger('Meraba1.0','Duration',-99);
  directry:=ayarini.ReadString('Meraba1.0','Directory','');
  rndm:=ayarini.ReadBool('Meraba1.0','Random',True);
  stret:=ayarini.ReadBool('Meraba1.0','Stretch',True);
  pws:=ayarini.ReadString('Meraba1.0','PWs','');
  if pws <>'' then pw:=ayarini.ReadBool('Meraba1.0','PW',True);

  DirectoryListBox1.FileList.Directory:=directry;
  TrackBar1.Position:=duration;
  checkbox1.Checked:=rndm;
  checkbox2.Checked:=stret;
  checkbox3.Checked:=pw;
  MaskEdit1.Text:=pws;

  ayarini.Free;
end;

procedure TForm2.AyarYaz;
var dizin:string;

```

```

    ayar:TIniFile;
begin
    dizin:=extractfilepath(application.ExeName);
    ayar:=TIniFile.Create(dizin+'Meraba.ini');
    ayar.WriteInteger('Meraba1.0','Duration',TrackBar1.Position);
    ayar.WriteString('Meraba1.0','Directory',DirectoryListBox1.FileList.Directory);
    ayar.WriteBool('Meraba1.0','Random',CheckBox1.Checked);
    ayar.WriteBool('Meraba1.0','Stretch',CheckBox2.Checked);
    if maskedit1.Text="" then CheckBox3.Checked:=false else CheckBox3.Checked:=true;
    ayar.WriteBool('Meraba1.0','PW',CheckBox3.Checked);
    ayar.WriteString('Meraba1.0','PWs',MaskEdit1.Text);
    ayar.Free;
end;

```

```

procedure TForm2.BitBtn2Click(Sender: TObject);
begin
    close;
end;

```

```

procedure TForm2.FormShow(Sender: TObject);
begin
    ayaroku;
end;

```

```

procedure TForm2.FormKeyDown(Sender: TObject; var Key: Word;
    Shift: TShiftState);
begin
    if key=VK_ESCAPE then close;
end;

```

```

procedure TForm2.CheckBox3Click(Sender: TObject);
begin
    MaskEdit1.Enabled:=checkbox3.Checked;
end;

```

```

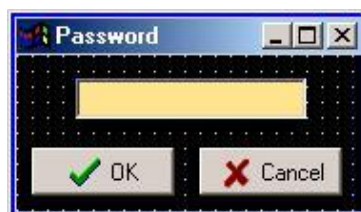
procedure TForm2.BitBtn3Click(Sender: TObject);
begin
    aboutbox.show;
end;

```

```

end.

```



Form3'ün kodları:

```

unit Unit3;

```

```

interface

```

```

uses

```

```

    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    StdCtrls, Buttons, Mask;

```

```

type

```

```

    TForm3 = class(TForm)

```

```

        BitBtn1: TBitBtn;

```

```

        BitBtn2: TBitBtn;

```

```

        MaskEdit1: TMaskEdit;

```

```

procedure BitBtn2Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
  htb:HWND;
  oldval:longint;
public
  { Public declarations }
end;

var
  Form3: TForm3;

implementation
uses Unit1;
{$R *.DFM}

procedure TForm3.BitBtn2Click(Sender: TObject);
begin
  close;
end;

procedure TForm3.BitBtn1Click(Sender: TObject);
var a,b:string;
begin
  a:=form1.maskedit1.text;
  b:=maskedit1.text;
  if b=a then begin
    ShowWindow(htb,SW_SHOW);
    SystemParametersInfo(97,word(false),@oldval,0);
    application.Terminate;
  end;
end;

procedure TForm3.FormShow(Sender: TObject);
begin
  htb:=FindWindow('Shell_Traywnd',nil);
  form1.Cursor:=crDefault;
  MaskEdit1.Text:="";
  maskedit1.SetFocus;
end;

procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  form1.Cursor:=crnone;
end;

end.

```



About – hakkında kısmının kodu:

```

unit About;
interface
uses Windows, Classes, Graphics, Forms, Controls, StdCtrls,
    Buttons, ExtCtrls, SysUtils;
type
TAboutBox = class(TForm)
    Comments: TLabel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    OKButton: TButton;
    Panel1: TPanel;
    Panel2: TPanel;
    ProductName: TLabel;
    procedure OKButtonClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Label1MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
    procedure Label2MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
    procedure Panel1MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
    procedure Label2Click(Sender: TObject);
    procedure Label1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    AboutBox: TAboutBox;
    stop: boolean;
implementation
{$R *.DFM}

procedure TAboutBox.OKButtonClick(Sender: TObject);
begin close; end;

procedure TAboutBox.FormCreate(Sender: TObject);
var MS: TMemoryStatus;
begin
    Application.ProcessMessages; GlobalMemoryStatus(MS);
    label5.Caption := FormatFloat('#,###" KB"', MS.dwTotalPhys / 1024);
    label6.Caption := Format('%d %%', [MS.dwMemoryLoad]);
end;

procedure TAboutBox.Label1MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin label1.Font.Color:=clblue; end;

procedure TAboutBox.Label2MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin label2.Font.Color:=clblue; end;

procedure TAboutBox.Panel1MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin label1.Font.Color:=clMaroon; label2.Font.Color:=clMaroon; end;

procedure TAboutBox.Label2Click(Sender: TObject);
begin
    winexec(pchar('start http://www.tbagriyanik.com'),0);
end;

procedure TAboutBox.Label1Click(Sender: TObject);
begin
    winexec(pchar('start mailto:tarik@tbagriyanik.com'),0); //mailto: Outlook ile mail atmamızı sağlar

```

```
end;  
end.
```

Şifreleme için basit bir yöntem: (Ornekler\Delphi\Meraba\sifre\Project1.dpr)

```
procedure TForm1.Button1Click(Sender: TObject);  
var i:integer;kelime:string;  
begin  
kelime:=edit1.text;  
for i:=1 to Length (kelime) do kelime[i]:=chr(ord(kelime[i])+i); //Toplama ile anlamsızlaştırılıyor  
label1.caption:=kelime;  
end;  
  
procedure TForm1.Button2Click(Sender: TObject);  
var i:integer;kelime:string;  
begin  
kelime:=label1.Caption;  
for i:=1 to Length (kelime) do kelime[i]:=chr(ord(kelime[i])-i); //Çıkarma ile eski haline geliyor  
label2.caption:=kelime;  
end;
```

Ord(tek_harf) ve Chr(ascii_deger) fonksiyonları sadece char tipli yani tek karakterlik işlemlerde kullanılır. Eğer bir kelimenin tümünü sayısal değere dönüştürmek veya bir sayıyı metin / alfa nümerik türe dönüştürmek isteseydik StrToInt, IntToStr, StrToFloat veya FloatToStr gibi fonksiyonlar kullanırdık. Tür dönüşümleri ile ilgili Delphi yardımına bakabilirsiniz.

Şifrelemenin daha gelişmiş yapıları da var. AND, OR, XOR, kaydırma, yer değiştirme, bu yöntemlerin karma kullanımı, belli bir anahtar ile maskeleye gibi yapılabilir. Asla Random (rasgele) bir değeri anahtar olarak kullanamayız. Çünkü daha sonra eski değerine döndürülmesi imkansız olur. Ters algoritma kullanılarak eski değere aynen ulaşılabilir. Fakat kolay algoritmaları deneme yanılma yöntemi ile çözmek kolaydır. Örneğin bizim örnekte kelime yerine 123 gibi bir rakam girildiğinde 246 elde edilmesi, şifre kırıcılar için çözülmesi gerçekten de kolay bir yöntem olduğu görülebilir. Encrypt (*enkript diye okunur* – şifreleme / encode) edilince aynı boyutta şifre elde etmek sakıncalı. Gerçek değer içine basamaklar arasına rasgele şaşırtıcı değerler sokulabilir. Kriptonun tersine decrypt (*dekript diye okunur* – şifre çözme / decode) denilir.

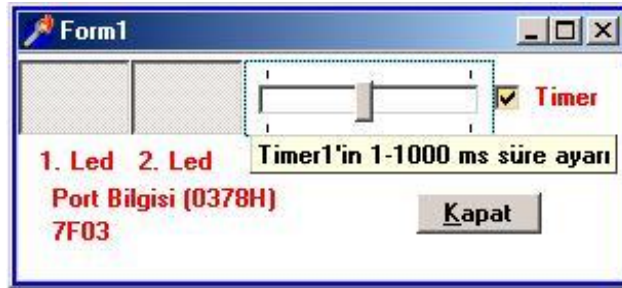
Başka bir ekran koruyucu örneği (ekrank.zip):



7. ParalelPort (Win9x için paralel veriyolu kullanımı)

(Ornekler\Delphi\ParalelPort\ledder.dpr)

Yazıcı bağlantı noktası olan LPT1 – Line Printer’i bi-directional – çift yönlü yani veri yollamak veya almak için kullanabiliriz. Adı üstünde paralel bağlantı imkanı olduğu için biz de aynı anda iki led – light-emitting diode kullanabiliyoruz. İsterseniz daha fazla bağlantı da yapabilirsiniz. Biraz da elektronik bilginiz varsa rölelerle ev aletlerini kontrol edebilirsiniz. Tüm pinler TTL – Transistor Transistor Logic seviyesindedir. Yüksek seviye 2.4V ile 5 V arası iken, düşük seviye 0V ile 0.8V arası kabul edilir. Maksimum 2.6 mA akım verebilir. Dikkatli olun, paralel port led yakabiliyor ama daha üst akım ve voltajlara dayanamaz. Bir buffer - tampon devresi gerekebilir. NT/2000 ve XP için bu anlatılanlar geçerli değildir. Ama XP’nin Compatibility – Uyumluluk modu sayesinde çalıştırılabilir. Örnek sonundaki açıklamaları okuyunuz. “Ornekler\Delphi\ParalelPort\trafik\Project1.dpr” örneğinde de bu uygulamanın bir benzeri olan iki (yeşil), üç (sarı) ve dördüncü (kırmızı) çıkış pinlerini kullanan, trafik ışıkları olarak görebilirsiniz.



Resim 12 – Paralel port ile led kullanma

Form nesneleri:

```
object Form1: TForm1
  Left = 492
  Top = 111
  Width = 305
  Height = 139
  Caption = 'Form1'
  Color = clWhite
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clRed
  Font.Height = -10
  Font.Name = 'MS Sans Serif'
  Font.Style = [fsBold]
  OldCreateOrder = True
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
    Left = 16
    Top = 64
    Width = 110
    Height = 13
    Caption = 'Port Bilgisi (0378H)'
  end
  object Label2: TLabel
    Left = 16
    Top = 80
    Width = 8
    Height = 13
    Caption = '0'
  end
  object Label3: TLabel
    Left = 10
    Top = 46
    Width = 37
    Height = 13
    Caption = '1. Led'
  end
```

```
end
  object Label4: TLabel
    Left = 58
    Top = 46
    Width = 37
    Height = 13
    Caption = '2. Led'
  end
  object Button3: TButton
    Left = 194
    Top = 68
    Width = 61
    Height = 20
    Caption = '&Kapat'
    TabOrder = 0
   OnClick = Button3Click
  end
  object ToolBar1: TToolBar
    Left = 0
    Top = 0
    Width = 295
    Height = 41
    ButtonHeight = 37
    ButtonWidth = 55
    Caption = 'ToolBar1'
    TabOrder = 1
    object ToolButton1: TToolButton
      Left = 0
      Top = 2
      Caption = '1'
      ImageIndex = 0
      Style = tbsCheck
      OnClick = ToolButton1Click
    end
    object ToolButton2: TToolButton
```

```

Left = 55
Top = 2
Caption = '2'
ImageIndex = 1
Style = tbsCheck
OnClick = ToolButton1Click
end
object TrackBar1: TTrackBar
Left = 110
Top = 2
Width = 122
Height = 37
Hint = 'Timer1'#39'in 1-1000 ms süre ayarı'
Max = 1000
Min = 1
Orientation = trHorizontal
ParentShowHint = False
Frequency = 1
Position = 1
SelEnd = 0
SelStart = 0
ShowHint = True
TabOrder = 0

```

```

TickMarks = tmBoth
TickStyle = tsManual
OnChange = TrackBar1Change
end
object CheckBox1: TCheckBox
Left = 232
Top = 2
Width = 57
Height = 37
Caption = 'Timer'
TabOrder = 1
OnClick = CheckBox1Click
end
end
object Timer1: TTimer
Enabled = False
Interval = 10
OnTimer = Timer1Timer
Left = 152
Top = 56
end
end

```

Unit1.pas yani asıl programın kodları:

```

unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  StdCtrls, ComCtrls, ExtCtrls, ToolWin;

type
  TForm1 = class(TForm)
    Button3: TButton;
    CheckBox1: TCheckBox;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Timer1: TTimer;
    ToolBar1: TToolBar;
    ToolButton1: TToolButton;
    ToolButton2: TToolButton;
    TrackBar1: TTrackBar;
    procedure Button3Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure TrackBar1Change(Sender: TObject);
    procedure ToolButton1Click(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  degis:boolean=true;

implementation

{$R *.DFM}

```

```

procedure TForm1.ToolButton1Click(Sender: TObject);
var deger:word;
begin
  if ToolButton1.Down and ToolButton2.Down then
    begin
      asm
        mov dx,0378h
        mov ax,03h //yani 0011
        out dx,ax
        mov dx,0378h
        in ax,dx
        mov deger,ax
      end;
    end;

  if (ToolButton1.Down=false) and
  (ToolButton2.Down=false)then
    begin
      asm
        mov dx,0378h
        mov ax,0h //yani 0000
        out dx,ax
        mov dx,0378h
        in ax,dx
        mov deger,ax
      end;
    end;

  if (ToolButton1.Down) and (ToolButton2.Down=false)then
    begin
      asm
        mov dx,0378h
        mov ax,01h //yani 0001
        out dx,ax
        mov dx,0378h
        in ax,dx
        mov deger,ax
      end;
    end;
  end;

```

```

end;

if (ToolButton2.Down) and (ToolButton1.Down=false)then
begin
asm
  mov dx,0378h
  mov ax,2h //yani 0010
  out dx,ax
  mov dx,0378h
  in ax,dx
  mov deger,ax
end;
end;
label2.Caption:=IntToHex(deger,4);
end;

procedure TForm1.CheckBox1Click(Sender: TObject);
begin
timer1.enabled:=checkbox1.Checked;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
asm
  mov dx,0378h
  mov ax,0h
  out dx,ax
end;
end;

```

```

close;
end;

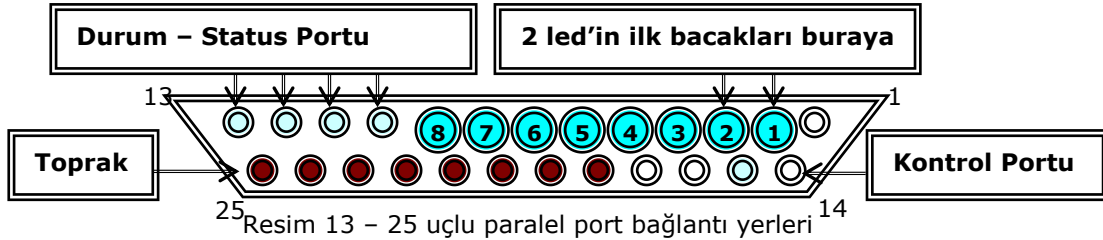
procedure TForm1.Timer1Timer(Sender: TObject);
begin
if degis then begin
asm
  mov dx,0378h
  mov ax,02h
  out dx,ax
end;
degis:=not(degis);
end else begin
asm
  mov dx,0378h
  mov ax,01h
  out dx,ax
end;
degis:=not(degis);
end;
end;

procedure TForm1.TrackBar1Change(Sender: TObject);
begin
timer1.Interval:=TrackBar1.Position;
end;

end.

```

Paralel port'un base – taban adresi 0378H. "Bilgisayarım*Denetim Masası*Sistem" 'den yazıcı bağlantı noktanızın Kaynaklar kısmında bu Giriş/Çıkış değeri görebilirsiniz. Aynı yolu izleyerek klavye 0060H, birinci seri bağlantı noktası 03F8H ve ikinci seri port 02F8H adresine sahip olduğunu görebilirsiniz. XP için ise "System*Hardware*Device Manager" ile donanım listesine ulaşabilirsiniz.



Paralel Bağlantı Noktası Bacak İşlevleri:

1	Strobe	Çıkış	14	Otomatik besleme – Autofeed	Çıkış
2	Veri biti 1	Çıkış	15	Hata – Error	Giriş
3	Veri biti 2	Çıkış	16	Sıfırla – Printer Reset	Çıkış
4	Veri biti 3	Çıkış	17	Select in	Çıkış
5	Veri biti 4	Çıkış	18	Toprak – Ground	
6	Veri biti 5	Çıkış	19	Toprak – Ground	
7	Veri biti 6	Çıkış	20	Toprak – Ground	
8	Veri biti 7	Çıkış	21	Toprak – Ground	
9	Veri biti 8	Çıkış	22	Toprak – Ground	
10	Bilgi – Acknowledge	Giriş	23	Toprak – Ground	
11	Meşgul – Busy	Giriş	24	Toprak – Ground	
12	Kağıt bitti – Paper Out	Giriş	25	Toprak – Ground	
13	Select Out	Giriş			

Yazıcıya bir belge yollanınca şu adımlar gerçekleşir: Strobe hattına en az 0.5 ms - mikrosaniye alçak darbe yollanır, Autofeed hattına alçak sinyali verilir, Init hattına en az 50 ms alçak sinyal yollanır, yazıcıyı seçmek için Select'e alçak sinyali yollanır, kesme kabul edilince Acknowledge hattı 0.5 ms kadar alçak darbesi taşır, yazma başlayacağı için Busy yükselir, sayfa sonlarında Paper Out (Paper End) hattı yükselir, hata anlarında Error yazıcı tarafından alçağa çekilir, yazıcı seçili olunca Select in yükseğe çekilir.

Tabloda görüldüğü gibi on yedi sinyal hattının sekizi veri, dördü kontrol, beşi de durum hattı olarak kullanılır. On iki veri çıkışı var iken beş hattı da giriş için kullanabiliriz. Led bacakları resimdeki bir ve ikinci noktalara takılır, diğer toprak bacakları istenirse kasaya değdirilebilir, istenirse aynı resimde gördüğünüz kahverengi renkli noktalara da takılabilir. İki led'in de birbirine değmemesini sağlayın. Eğer yanmıyorlarsa yönlerini değiştirin.

Programın mantığı 2'den fazla led için pek uygun değil. Çünkü 2 led için $2^2=4$ ihtimal olduğu için if - eğer bloğundan dört adet kullanıyorken, örneğin sekiz led için, $2^8=256$ ihtimal yani 256 if bloğu kullanılamaz. Bu arada aklıma Assembler'de "and" ve "or" komutları ile maskeleme kullanımı geliyor, yani ilk led'in durumu ne olursa olsun ikinci led aç kapa yapabilmeniz ancak bu şekilde olabilir.

<u>x86 Assembly</u>	<u>Borland C</u>	<u>Microsoft C</u>
out port#, AL	outportb(port#, byte)	outp(port#, byte)
in AL, port#	inportb(port#)	inp(port#)

SPP – Standart Parallel Port : ISA uyumlu, AT tipli tek yönlü standart.

EPP – Enhanced Parallel Port : Geliştirilmiş daha hızlı paralel port (1 - 2 MB arası). PS/2 portunu emüle edebilen, paralel port ağı oluşturabilen çift yönlü bağlantı noktası.

ECP – Extended Parallel Port : Genişletilmiş SPP – Standart Parallel Port yani Centronics modunun gelişmiş hali. ISA hızında aktarım yapabilir. DMA yeteneği sayesinde çift yönlü çalışabilir.

"Başlat*Çalıştır...*debug" komutu ile kurulu olan LPT port'larının listesini elde edebiliriz:

```
-d 0040:0008 L4
0040:0000          78 03 00 00          x...
```

Hafızadaki bu bilgilere göre 1 adet yazıcı portumuz 0378H adresinde. İkincisi olsaydı 00 00 yazan yerde 78 02 gibi bir rakam olmalıydı. Tabii ki "Bilgisayarım*Denetim Masası*Sistem*Aygıt Yöneticisi" 'ne gelip, Bilgisayar simgesi çift tıklanıp, Giriş / Çıkış (G/Ç veya Input / Output - I/O) işaretlenerek de kim hangi adresi kullanıyor görebilirsiniz.

0378H için veri girişini, durum portu olan 0379H ve kontrol portu olan 037AH adreslerini programlayıp elde edebiliriz. C dilindeki örnekte 0379H'nin ilk biti giriş için kullanılıyor.

İsterseniz Delphi ve assembler dillerini veri giriş ve çıkışı için daha modüler kullanalım:

```
procedure PortOut(IOPort:word; value:byte);
assembler;
asm
xchg ax,dx
out dx,al
end;
```

```
function PortIn(IOPort:word):byte;
assembler;
asm
mov dx,ax
in al,dx
end;
```

Tek yapacağımız bir porta veri yollamak için
PortOut (portadres, deger);
 veri okumak için
deger := PortIn (portadres);

Pascal dilinde hazır Port komutu kullanarak da paralel port'ta dört adet ışık kaydırabiliriz:

```
uses crt;
var
```

```

    lambalar : byte;
begin
    lambalar:=1;
    repeat
        port[$378]:=lambalar;
        lambalar:=(lambalar shl 1) mod 16;
        delay(500);
    until keypressed;
end.

```

C dilinde basit bir veri giriş çıkış örneği: (Ornekler\Delphi\ParalelPort\Led\led.c)

```

/* led.c*/
#include <stdio.h>
#include <dos.h>          /*delay() komutu için*/

void main(void)
{
    int in;

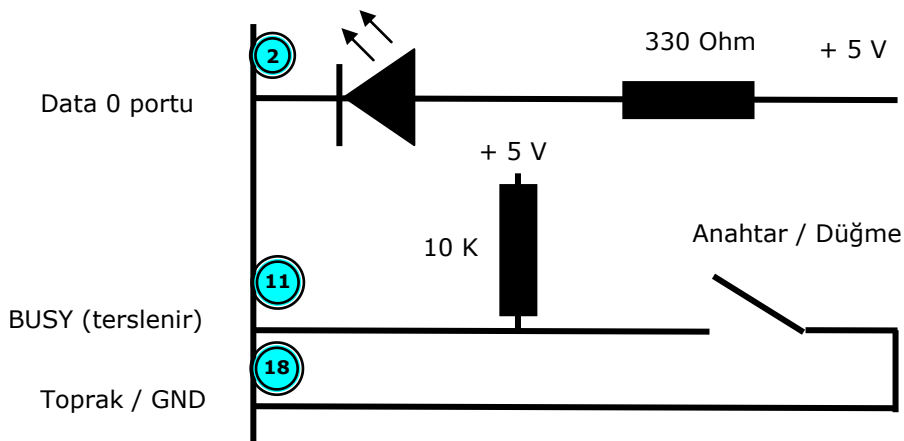
    while(1)
    {
        in=inportb(0x0378 + 1);

        if (((in^0x80) & 0x80) == 0) //düğme basılı mı? 1000 0000b ile data maskeleyme
        {
            outportb(0x0378,0x00); /* led açık ON*/
            delay(100);

            outportb(0x0378,0x01); /* led kapalı OFF*/
            delay(100);
        }
        else
        {
            outportb(0x0378,0x01); /* led kapalı OFF*/
        }
    }
}

```

Üstteki led.c programımız için devremizde iki direnç, birer adet Led ve anahtar bulunuyor. 330 Ω , 10 K Ω direnç değerleridir.



Resim 14 – Paralel port veri okuma ve yollama devresi

Seri portumuz olan COM – communication (iletişim) portlarından iletişim ise daha basit gibi gözükse de, aslında paralel porta göre daha kolay değildir. Akbil ve TOM (Touch Memory) gibi aygıtlar seri iletişimle veri alışverişinde bulunuyor. RS-232 standardına göre hat sayısı 20 olmasına rağmen, bize sadece TXD, RXD ve GND toprak hattı yeterlidir. Üç hat ile çalışan kablolar Null (boş) Modem kablosu veya cross (çapraz) kablo da denir. Diğer hatlar ise güvenlik ve daha hızlı veri transferi için kullanılıyor. Fare veya dış modem gibi aletleri bilgisayarımızın

arkasındaki bu 9 iğneli portlara takabiliriz. Bilgisayarımız 16550 adlı 115200 bps (bits per second) hızına imkan veren UART – Universal Asynchronous Receiver Transmitter devreleri ile dış dünya ile veri alışverişinde bulunur. Aslında bu çip seri iletişimi paralele çevirir. COM1 için 03F8H – 03FFH, COM2 için de 02F8H – 02FFH adresleri aralığını kullanabiliriz. Veriler paket halinde peş peşe yollandığı için veri göndermeden önce başlangıç biti, 5 veya 8 bit veriden sonra da bitiş biti ve parite biti yollar, bu bitlerin sinyal değerleri +12V ve –12V arasındadır. Bu arada başka bir kaynakta da değer aralığı olarak –3 V ile +25 V arası olduğunu gördüm. Kızılötesi (infrared) iletişim de ancak seri olabilir. Modem'in anlamı da modülatör, demodülatör yani sese çevirip tekrar dijital bilgiye çevirebilmedir, ki bu işlemler seri iletişim sayesinde gerçekleşir. Veri akışı bir saat sinyali ile kontrol edilir. Sinyali alan ile yollayan aynı saat darbesine (senkron iletişim) sahip olmalıdır. Eğer başlangıç ve son bit şeklinde veriler paketlenerek yollanıyorsa asenkron iletişim yapılıyor demektir. Modemin çalışırken çıkardığı garip seslerin sebebi de iki iletişim biriminin anlaşmaya çalışmasıdır. Belli bir protokol birliğine de varınca iletişim devam eder. Tek yönlü iletişime half duplex – yarı iki yön, her iki yönde iletişime de full duplex – tam iki yön denir.

Basic dilinde COM1'e veri yollamak için şu komutu kullanabiliriz:

Open “com1:300,e,7,1” as #1

'300 baud hızında, parite even – çift, 7 bit veri, 1 stop – bitiş biti

Seri Bağlantı Noktası Bacak İşlevleri:

1	Carrier Detect	6	Data Set Ready
2	Bilgi Alma – Receive Data	7	Request to Send
3	Bilgi Yollama – Transmit Data	8	Clear to Send
4	Data Terminal Ready	9	Zil - Ring Indication
5	Toprak – Ground		

Biraz konudan kayıyoruz ama benzer bir konu olarak, bilgisayarımızın speaker – hoparlörünü kullanarak basitçe ses çıkarabiliriz: (Ornekler\C\c.zip*speaker.c)

```
#include <dos.h>
#include <conio.h>
#define spkport 0x61
main()
{
    unsigned char port;
    port= inportb (spkport) & 0xFC; //0xFC = 1111 1100b
    while (!kbhit())
    {
        outportb (spkport, port | 2);
        delay (1);
        outportb (spkport, port);
        delay (1); }
}
```

Daha güzel bir örnek, PIT – Programmable Interval Timer kullanımı: (Ornekler\C\speaker2.c)

```
#include <dos.h>
#include <conio.h>
#define spkport 0x61
#define pitkomut 0x43
#define pitsyc2 0x42
#define pitsaatfark 0x1234DD
typedef unsigned char byte;
typedef unsigned short word;

void spksound(word frekans)
{
    word sayac;
    sayac= (int) pitsaatfark / frekans;
    outport (pitkomut , 0xB6); //0xB6 = 1011 0110b
    outport (pitsyc2 , (byte) sayac & 255); //LSB
    outport (pitsyc2 , (byte) (sayac >> 8)); //MSB
    outport (spkport , inportb (spkport) | 3); // xxxx xx11 maskesi
}
```

```
void spkNosound()
{ output (spkport , inportb (spkport) & 0xFC); // 1111 1100
}
```

```
main()
{word i;
for (i=20; i<=1500; i++)
{
    spksound(i);
    delay(1);
}
spkNosound();
return 0;
}
```

Eh, basit bip sesi çıkardıktan sonra *.wav dosya da speaker kullanarak dinleyebiliriz: (Ornekler\C\speaker3.c)

```
#include <dos.h>
#include <conio.h>
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#define spkport 0x61
#define pitkomut 0x43
#define pitsyc2 0x42
#define pitsaatfark 0x1234DD
typedef unsigned char byte;
typedef unsigned short word;
long beklemesuresi = 1200;

void sesiCal(byte *ses, word uzunluk)
{byte sayacdeger[256];
word i,bekle;

for (i=0; i<256; i++) sayacdeger[i] = i * 54 /255;

output (pitkomut , 0x90); // 0x90 = 1001 0000b
output (spkport , inportb (spkport) | 0x03); // xxxx xx11 maskesi
asm cli; // kesmeleri kapa

for (i=0; i<uzunluk; i++)
{
    output (pitsyc2 , sayacdeger[*ses]);
    for (bekle=0; bekle<=beklemesuresi; bekle++);
    output (pitsyc2 , sayacdeger[*ses]);
    for (bekle=0; bekle<=beklemesuresi; bekle++);
    ses++;
}
asm sti; // kesmeleri ac
output (spkport , inportb (spkport) | 0xFC);
output (pitkomut , 0xB6);
}

main(int psayisi, char *parametre[ ])
{FILE *wavdosya;
long wavuzunlugu;
byte *wavs;

if (psayisi==3) beklemesuresi = atoi (parametre[2]);
wavdosya = fopen(parametre[1], "rb");
wavuzunlugu = filelength (fileno(wavdosya));
if (!wavdosya) {printf("Dosya bulunamadi...");exit(1);}
if (wavuzunlugu>60000) wavuzunlugu=60000;
wavs = (byte *) malloc (wavuzunlugu);
if (!wavs) {printf("Bellek yetersiz...");exit(1);}
fread (wavs, 1, wavuzunlugu, wavdosya);
sesiCal (wavs, wavuzunlugu);
```

```

free (waves);
fclose (wavdosya);
return 0;
}

```

"c:\>speaker3 sesdosyasi.wav 5000" gibi kullanımı var. Hız ile ilgili olan 5000 yerine 600 ile 5000 arası bir rakam ile ses daha net oluncaya dek deneyebilirsiniz. Gerçekte speaker sadece 1 ve 0 ile çalışırken biz *.wav dosyasını 8 bit bilgilerini kullanarak sample – belli bir andaki ses örneği halinde dinliyoruz. Parametrik kullanımdan dolayı, isterseniz bir *.wav dosyayı programınız üzerine fare ile sürükleyerek de deneyebilirsiniz. Kalite berbat ama bazı ses dosyalarında hemen hemen iyi duyulabiliyor.

Genellikle ses kartının üzerinde bulunan 15 pinlik Game Port (Midi Port) yani joystick portunu da programlayabiliriz. Fakat seri port konusu gibi fazla detaya girmeden bu veri yolunu size tanıtayım. Donanım olarak joystick üzerinde bilgisayarımıza analog veri yollayan iki potansiyometre – değişken değerli direnç vardır. Interrupt 15H fonksiyon 01H, hafızada ise 0200H ve 0207H adres aralığını kullanır. Tuş basımlarını da 00H fonksiyonu ile AL yazmacından okuyabilirsiniz. Pot'lar x ve y eksen bilgilerini +5 volt üzerinden ADC – Analog to Digital Convertor devresine verir ve bu değer 0 - 255 değerleri arası olan 8 bit seviyesine iner. Dış dünya analog olduğuna göre ısı, ses, basınç ve ağırlık gibi bilgileri ADC devreleri ile bilgisayarımıza aktarabiliriz. Ses kartları bu işlemin tersi olan DAC – Digital to Analog Convertor devresi ile ses sentezi yaparlar. Genellikle ya sekiz ya da on altı bit ses kaydı yapılabilir. Monitörlerimize de RGB – Red Green Blue bilgileri DAC devreleri ile oluşturulur.

Oyun Bağlantı Noktası Bacak İşlevleri:

1	+5 V DC	9	+5 V DC
2	Çubuk 1, düğme 1	10	Çubuk 2, düğme 1
3	Çubuk 1 x değeri	11	Çubuk 2 x değeri
4	Toprak – Ground	12	Toprak – Ground
5	Toprak – Ground	13	Çubuk 2 y değeri
6	Çubuk 1 y değeri	14	Çubuk 2, düğme 2
7	Çubuk 1, düğme 2	15	+5 V DC
8	+5 V DC		

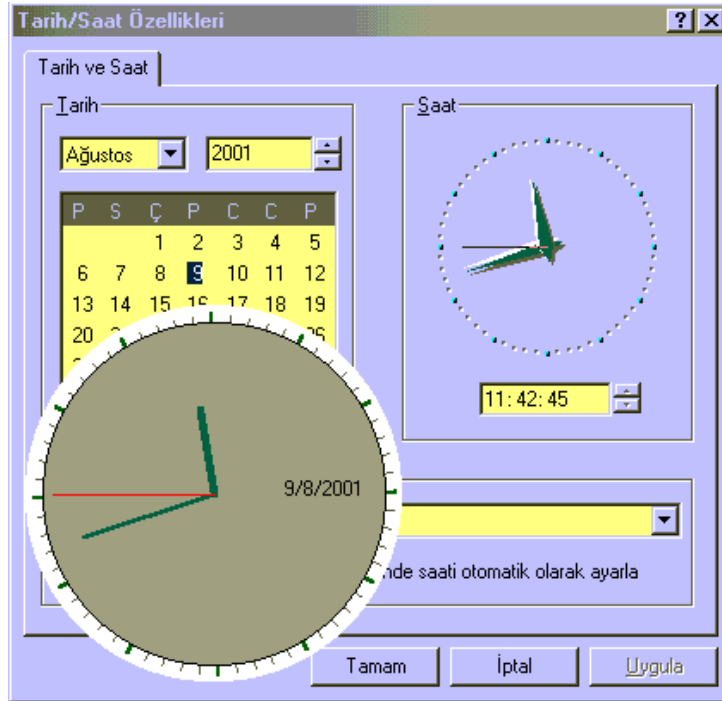
Windows içinde üç şekilde donanıma (sabit disk, ses kartı, bağlantı noktaları) erişilebilir: doğrudan giriş çıkış, aygıt sürücüler (*.*vxd ve Kernel modu) ve API kullanımı. Win 3.x/95/98/ME için doğrudan giriş çıkış komutları rahatlıkla kullanılabilirken, NT ve üzeri için bu imkansızdır. Win 3.x/95/98/ME işletim sistemlerinde çalışmasının sebebi, bu sistemlerin donanım ile yazılım arasında bir katmana sahip olmamasıdır. Ama aynı anda çalıştırılan bileşene başka program erişemez veya erişse bile sorunlar çıkabilir. VXD (Virtual Device Driver) ile Win9x, Kernel modu sürücüler ile de NT/2000/XP üzerinde erişim hakkı vardır. Windows API kullanarak da yapabiliriz fakat onun da dezavantajları tüm aygıtların standart olarak fonksiyonlarının olmaması ve port erişiminin yapılamamasıdır. XP işletim sisteminde *.exe dosyaya sağ tıklayıp, Compatibility – Uyumluluk sayfasında Windows 95, Windows 98 – ME, Windows NT – 2000 ile uyumlu çalışmalarını sağlayabiliriz. Bu sayede XP'nin hata verdiği komutları giderebiliyoruz.

Tabii ki gelişmiş ses kartlarımız, ekran kartlarımız varken bu zavallı bileşenleri neden programlıyoruz, diye sorarsanız aslında eskiden demo - demonstration veya intro - introduction üreten programcılar neler yaptıklarını görmek, diye bir cevap verebilirim. Delphi veya VB'de hazır nesneler ile ses dosyası çalmak veya yazıcıya bir bilgi yollamak çok daha zahmetsiz...

8. Saat (yuvarlak form ve çizim yapılması)

(Ornekler\Delphi\doner\Project1.dpr)

Sistem saati ve tarihini ekrana yuvarlak bir form içinde gösteren bir Delphi6 uygulamamız.



Resim 15 – Saat projemiz

Resimde görüldüğü gibi Windows'un Tarih/Saat menüsü ile birlikte, saatimizin değerleri karşılaştırabilirsiniz. Form üzerinde sadece Timer1, SavePictureDialog1 ve Image1 nesnesi bulunuyor.

object Form1: TForm1

Left = 271

Top = 104

BorderStyle = bsNone

Caption = 'Çiko'#39's Saat'

ClientHeight = 338

ClientWidth = 411

Color = clBtnFace

Font.Charset = DEFAULT_CHARSET

Font.Color = clWindowText

Font.Height = -11

Font.Name = 'MS Sans Serif'

Font.Style = []

OldCreateOrder = False

Position = poDesktopCenter

OnCreate = FormCreate

OnKeyPress = FormKeyPress

PixelsPerInch = 96

TextHeight = 13

object Image1: TImage

Left = 0

Top = 0

Width = 411

Height = 338

Align = alClient

OnMouseDown = Image1MouseDown

end

object Timer1: TTimer

Interval = 1000

OnTimer = Timer1Timer

Left = 48

Top = 56

end

object SavePictureDialog1: TSavePictureDialog

Filter = 'Bitmaps (*.bmp)|*.bmp'

Left = 120

Top = 56

end

end

Kodlarımız ise:

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, StdCtrls, Buttons, [math](#), jpeg;

type

```
TTip = (FSekizGen, FDaire, FDefault, FRoundKare, FSinCos);
TForm1 = class(TForm)
  Image1: TImage;
  Timer1: TTimer;
  procedure donder;
  procedure Timer1Timer(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  Procedure SetForm;
  procedure SetStyle(Value: Ttip);
  procedure Image1MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
  procedure FormKeyPress(Sender: TObject; var Key: Char);
```

private

```
{ Private declarations }
```

public

```
{ Public declarations }
```

end;

var

```
Form1: TForm1;
sonx,sony:integer;
FTip :Ttip;
R :HRgn;
```

implementation

```
{ $R *.DFM }
```

```
procedure tform1.donder;
```

```
var a,b:integer;
```

```
simdi: TDateTime;
Hour, Min, Sec, MSec: Word;
```

begin

```
simdi:=now;
DecodeTime(simdi, Hour, Min, Sec, MSec);
```

```
a:=round(sin(-(hour*30+min*0.5)*Pi/180)*50);
b:=round(cos(-(hour*30+min*0.5)*Pi/180)*50);
image1.Canvas.pen.color:=clteal;
image1.Canvas.moveTo(200,150);
image1.Canvas.pen.width:=4;
image1.Canvas.LineTo(200-a,150-b);
```

```
a:=round(sin(-min*6*Pi/180)*80);
b:=round(cos(-min*6*Pi/180)*80);
image1.Canvas.pen.color:=clteal;
image1.Canvas.moveTo(200,150);
image1.Canvas.pen.width:=2;
image1.Canvas.LineTo(200-a,150-b);
```

```
a:=round(sin(-sec*6*Pi/180)*95);
b:=round(cos(-sec*6*Pi/180)*95);
image1.Canvas.pen.color:=clred;
image1.Canvas.moveTo(200,150);
image1.Canvas.pen.width:=1;
image1.Canvas.LineTo(200-a,150-b);
```

end;

```
procedure TForm1.Timer1Timer(Sender: TObject);
```

```
var i,a,b:integer;
```

```
simdi: TDateTime;
Year, Month, Day: Word;
```

begin

```
simdi:= Now;
```

```

DecodeDate(simdi, Year, Month, Day);

for i:=0 to 60 do begin
  a:=round(sin(-(i*6-1)*Pi/180)*105);
  b:=round(cos(-(i*6-1)*Pi/180)*105);
  image1.Canvas.pen.width:=1;
  image1.Canvas.pen.color:=clgray;
  image1.Canvas.moveTo(200,150);
  image1.Canvas.LineTo(200-a,150-b);
end;
for i:=0 to 12 do begin
  a:=round(sin(-(i*30-1)*Pi/180)*105);
  b:=round(cos(-(i*30-1)*Pi/180)*105);
  image1.Canvas.pen.width:=2;
  image1.Canvas.pen.color:=clgreen;
  image1.Canvas.moveTo(200,150);
  image1.Canvas.LineTo(200-a,150-b);
end;
image1.Canvas.pen.width:=1;
image1.Canvas.pen.color:=clblack;
image1.Canvas.MoveTo(200,150);
image1.Canvas.Ellipse(100,50,300,250);
image1.Canvas.pen.width:=2;
image1.Canvas.pen.color:=clBlue;
image1.Canvas.TextOut(240,140,inttostr(day)+'/'+inttostr(month)+'/'+inttostr(year));
donder;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  SetStyle(FDaire);
  image1.Canvas.brush.Color:=clLtGray;
  SetWindowLong(Application.Handle, GWL_EXSTYLE, WS_EX_TOOLWINDOW); //Başlat çubuğunda gözükmez
end;

procedure TForm1.SetStyle(Value: Ttip);
begin
  FTip := value;
  SetForm;
end;

Procedure TForm1.SetForm;
begin
with form1 do begin
case ftip of
  FDaire:
    begin
      R:=CreateEllipticRgn(90,40,310,260);
      SetWindowRgn( Handle,R,True);
    end;
end;
Update;
end;
end;

procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
const
  SC_DragMove = $F012;
begin
  if button = mbright then
    begin
      SavePictureDialog1.Execute;
      if SavePictureDialog1.FileName<>'' then
        image1.Picture.SaveToFile(SavePictureDialog1.FileName);
    end
end

```



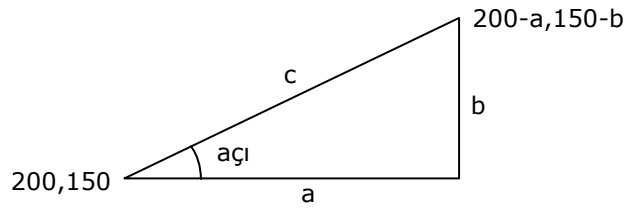
```

else begin
ReleaseCapture;
FORM1.perform(WM_SysCommand, SC_DragMove, 0);
end;
end;

procedure TForm1.FormKeyPress(Sender: TObject; var Key: Char);
begin if key=#27 then close; end;
end.

```

Image1'in Canvas - yüzey özelliğini Form1.Canvas'a göre tercih etmemizin sebebi, daha kalıcı etki elde etmemizdir. Eğer Form1.Canvas yapsaydık, form üzerinden bir pencere geçince yaptığımız şekiller silinecekti, yada saçma şekillerle karşılaşabilirdik. Windows her pencerenin üzerine resim olarak saklamıyor. 1280x1024x16 milyon renk çözünürlükte 4MB hafızayı bitmap görüntüye ayırmak büyük hafıza kaybı olurdu. Biz de istersek formumuzu sağ tıkladığımızda saatimizi *.bmp olarak kaydedebiliyoruz. Saniye veya diğer çizgilerin dönmesini trigonometri ile hallediyoruz. Timer1 her 1000 milisaniyede "donder" prosedürünü çağırıyor.



Resim 16 – Biraz trigonometri bilgisi

$a = \sin(\text{açı}) \cdot c$ x eksenindeki LineTo için gerekli değer
 $b = \cos(\text{açı}) \cdot c$ y eksenindeki LineTo için gerekli değer

Hipotenüs yani c, örneğin akrep için 50 piksel uzunluğunda olması gereken değerdir. c değeri her an aynı değerde olurken, a ve b kenarları her açı için ayrı değer kazanıyor. Öyleyse a ve b'yi hesaplamamız yeterlidir.

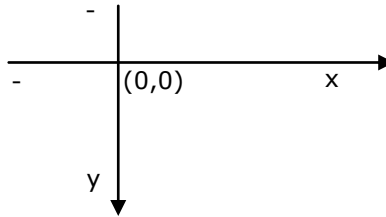
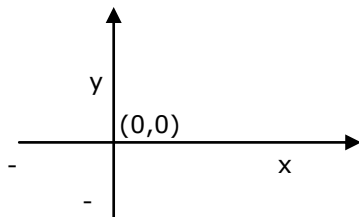
360 derecede 12 saat olduğu için, 360/12 sonucu olan 30° ile saati çarpıyoruz. Fakat dakikanın değeri saate eklenmesi için de, 30/60 yani 0.5° ile dakikayı çarpıp saate ekliyoruz. Dakika ve saniyenin toplam açısını bulmak için $360/60 = 6^\circ$ rakamı işimize yarıyor.

```

a:=round(sin(-min*6*Pi/180)*80);      //tamsayı x olması için round (real);
b:=round(cos(-min*6*Pi/180)*80);
image1.Canvas.pen.color:=clteal;
image1.Canvas.moveTo(200,150);      //çizgi başlangıç noktası (x,y) olarak
image1.Canvas.pen.width:=2;
image1.Canvas.lineTo(200-a,150-b);      //çizgi bitim koordinatı

```

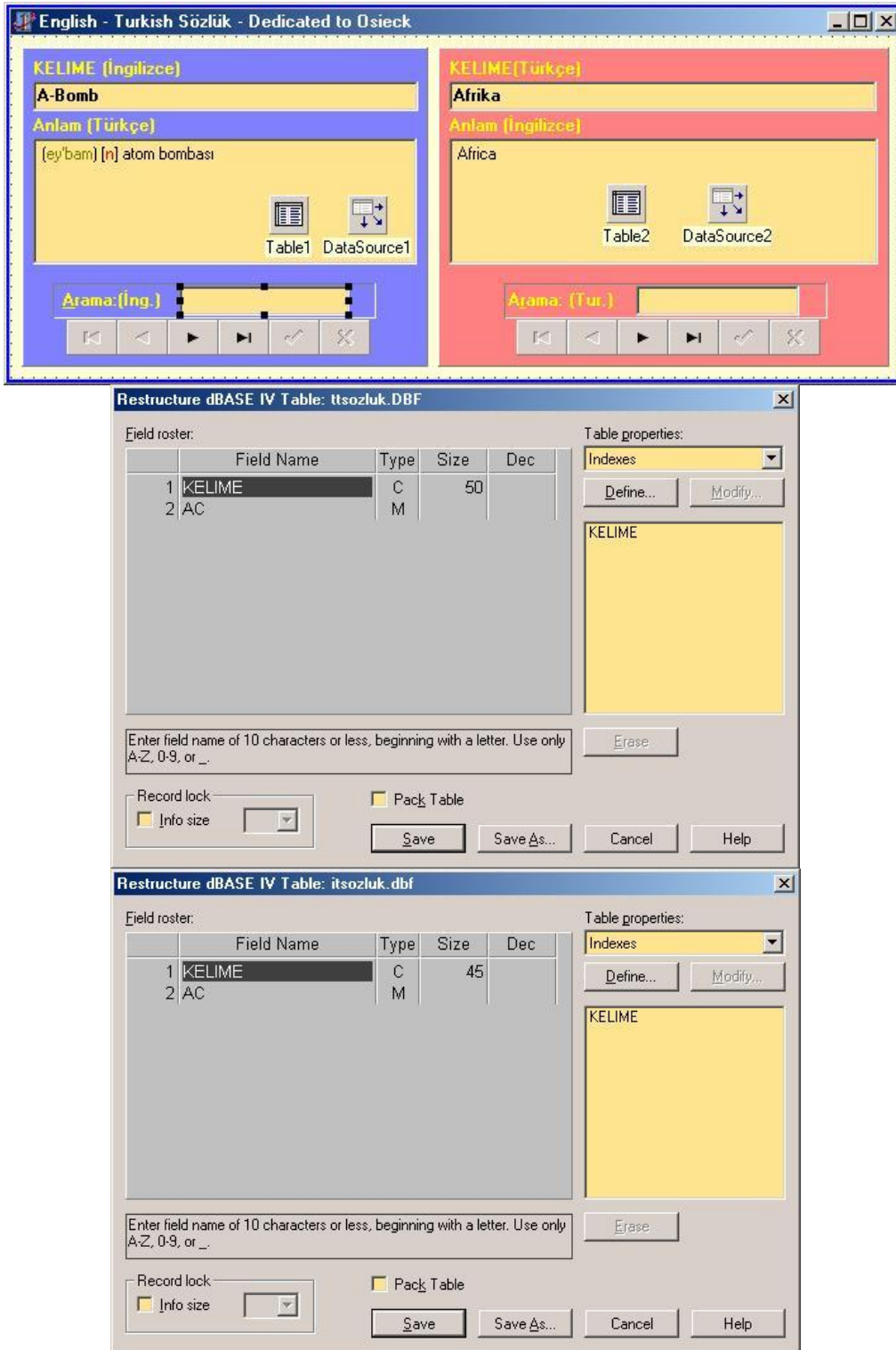
Sin ve cos fonksiyonlarının içini "-" negatif yaparak saat yönünde dönmesi sağlarız. Pi yani π aslında system unit'inde tanımlanmış bir fonksiyondur ve değeri 3.1415926535897932385'dir. $180^\circ = \pi$ olduğuna göre; $1^\circ = \pi / 180$ olur. Bu radyal dönüşümünü istediğimiz açıya uygulayarak, doğru sinüs ve kosinüs değerlerini elde ediyoruz. Bir çizgi hareket ettirilirken önceki eski çizgi silinmelidir. Fakat Canvas'ın üzerine çizdiğimiz elips sayesinde buna gerek kalmadı ve flicker yani titreme olmadan rahatlıkla Timer1 ile işlemlerimizi hallettik.



Matematikteki ve bilgisayardaki x - y düzlemi karşılaştırması, sadece y'de farklılık vardır.

9. Sozluk (İngilizce Türkçe sözlük)

(Ornekler\Delphi\Sozluk\Sozluk.dpr)



Resim 17 – Sözlük programı

Veritabanı kullanımı ile ilgili çok basit ve kullanışlı bir program. İki veritabanı dosyamız var. Kelime ve Ac alanlarından oluşan, biri İngilizce'den Türkçe'ye, diğeri de Türkçe'den İngilizce'ye anlam bilgilerini saklayan tablo. Bu tabloları bedava sözlük olan Dictionary On-Top programından aldım.

Programın kodları :

```

unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Db, DBTables, StdCtrls, ComCtrls, DBCtrls, Mask, ExtCtrls;

type
  TForm1 = class(TForm)
    Bevel1: TBevel;
    Bevel2: TBevel;
    DataSource1: TDataSource;
    DataSource2: TDataSource;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBNavigator1: TDBNavigator;
    DBNavigator2: TDBNavigator;
    DBRichEdit1: TDBRichEdit;
    DBRichEdit2: TDBRichEdit;
    Edit1: TEdit;
    Edit2: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Panel1: TPanel;
    Panel2: TPanel;
    Table1: TTable;
    Table1AC: TMemoField;
    Table1KELIME: TStringField;
    Table2: TTable;
    Table2AC: TMemoField;
    Table2KELIME: TStringField;
    procedure Edit1Change(Sender: TObject);
    procedure Edit2Change(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.DFM}

procedure TForm1.Edit1Change(Sender: TObject);
begin if edit1.Text<>'' then table1.locate('KELIME',edit1.text,[loCaseInsensitive, loPartialKey]); end;

procedure TForm1.Edit2Change(Sender: TObject);
begin if edit2.Text<>'' then table2.locate('KELIME',edit2.text,[loCaseInsensitive, loPartialKey]); end;

end.

```

Görüldüğü gibi sadece edit1 ve edit2 içine iki satır kod yazmışız. Amaç iki veritabanı içinden kelime alanına dayalı arama yapmak. Basit bir tasarım ile işlevsel bir sözlük elde ettik. Yeni kayıt yapılmasını DBNavigator nesnelerinde kapattım, aslında hiç gerek yok. Kullanıcı kendi sözcüklerini de sözlüğe ekleyebilir veya varolan kelimelerde düzenlemeler yapabilir. Karşılaştığım en büyük sorun dbase'e sözlüğü aktarırken Türkçe karakterlerin garipleşmesi idi. Biraz gayret ederek tablonun dilini Türkçe yapınca sorun çözüldü.

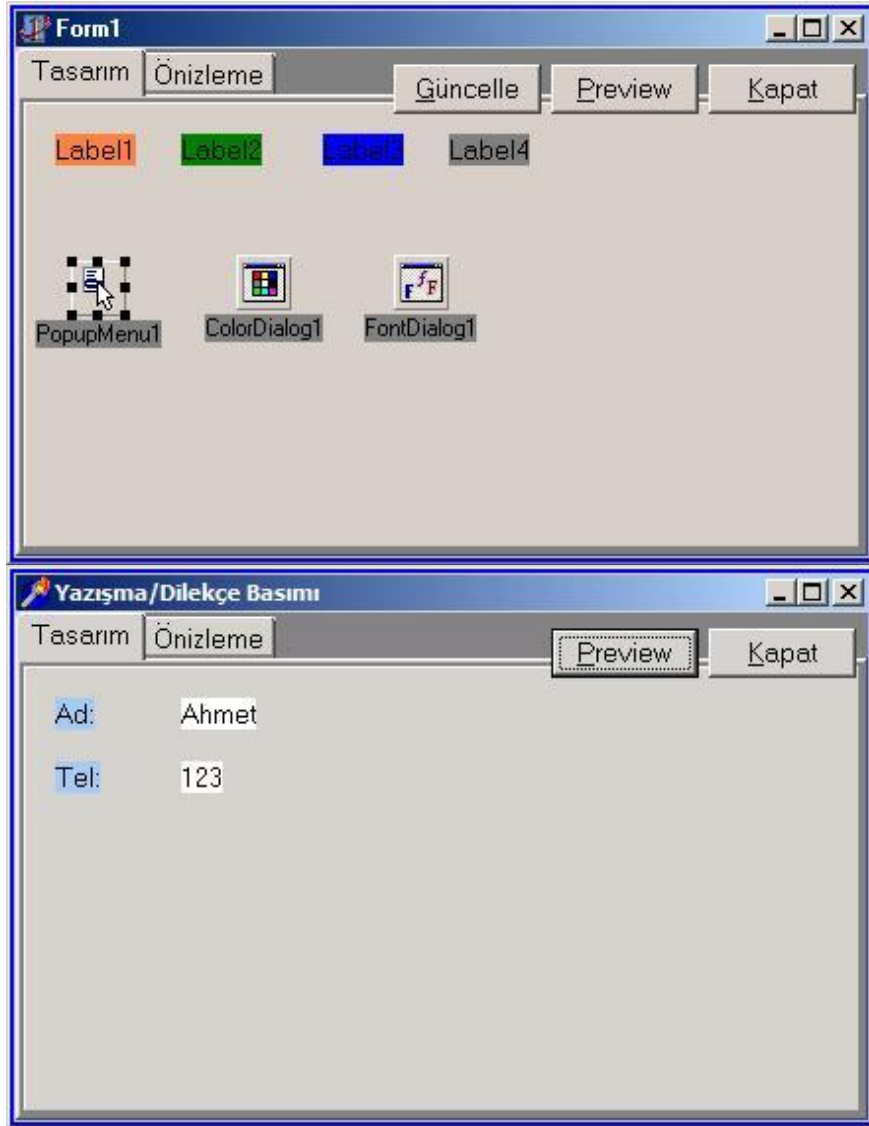
10. surukle (nesne sürüklenme ve yazıcı kullanımı)

(Ornekler\Delphi\surukle\Project1.dpr)

Bu projede asıl amacım yazıcı çıktısını çalışma anında özelleştirmekti. Şimdilik sadece Label – etiketler rahatça hareket ettirilebiliyor (drag and drop – sürükle ve bırak). İsterseniz kayıt ve yükleme gibi yapılan dökümü kaydetmeyi de ekleyebilirsiniz, sadece dört etiket yetmez deyip RadioButton, CheckBox, Button... da yapabilirsiniz, bu arada nesne create – oluşturmayı da deneyebilirsiniz. "Ornekler\Delphi\Dolphin\Dolphin.dpr" örneğinde de ListView nesnesinden StringGrid nesnesine sürükle bırak var. Programınıza dışarıdan dosya sürüklenmesi gibi kodları da DelphiTurk.com sitesinden elde edebilirsiniz.

Kod kısmı çok uzun gibi. "case" komutu için daha iyi bir çözüm aranabilir. Çünkü daha fazla nesne eklendiğinde program gereksiz yere uzayacak. Label'lara sağ tıklandığında "Yazı Tipi" ve "Renk" seçimleri çıkacak olan bir PopupMenu koyuyoruz. Yazı tipi FontDialog1'i, Renk de ColorDialog1'i çalıştırıyor.

Önizleme sayfasına bir adet QuickRep ve dört adet de QRLabel koymanız yeterli. Preview düğmesi ile QuickRep'i önizleme yapabiliyoruz. Delphi5 ve üzeri kullanıyorsanız QuickRep veritabanına bağlı değilse önizleme boş çıkabilir. Ayrıca QuickRep nesnesi üzerine QRLabel'lerinizi boş bulduğunuz yere koymayın, "Bands*HasColumnHeader" i True yaparak başlık kısmına koyabilirsiniz.



Resim 18 – Tasarım ve çalışma anı görüntüleri

Kodlar:

```

unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, quickrpt, QrCtrls, ComCtrls, Menus;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    ColorDialog1: TColorDialog;
    FontDialog1: TFontDialog;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    PageControl1: TPageControl;
    Panel1: TPanel;
    PopupMenu1: TPopupMenu;
    QRLabel1: TQRLabel;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QuickRep1: TQuickRep;
    Renk1: TMenuItem;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    YazTipi1: TMenuItem;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Panel1DragDrop(Sender, Source: TObject; X,
      Y: Integer);
    procedure Panel1DragOver(Sender, Source: TObject; X,
      Y: Integer;
        State: TDragState; var Accept: Boolean);
    procedure Label3MouseDown(Sender: TObject; Button:
      TMouseButton;
        Shift: TShiftState; X, Y: Integer);
    procedure Button3Click(Sender: TObject);
    procedure Label4MouseDown(Sender: TObject; Button:
      TMouseButton;
        Shift: TShiftState; X, Y: Integer);
    procedure Label1MouseDown(Sender: TObject; Button:
      TMouseButton;
        Shift: TShiftState; X, Y: Integer);
    procedure Label2MouseDown(Sender: TObject; Button:
      TMouseButton;
        Shift: TShiftState; X, Y: Integer);
    procedure Label1StartDrag(Sender: TObject;
      var DragObject: TDragObject);
    procedure Label2StartDrag(Sender: TObject;
      var DragObject: TDragObject);
    procedure Label3StartDrag(Sender: TObject;
      var DragObject: TDragObject);
    procedure Label4StartDrag(Sender: TObject;
      var DragObject: TDragObject);
    procedure YazTipi1Click(Sender: TObject);
    procedure Renk1Click(Sender: TObject);
  private
    { Private declarations }

```

```

  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  suruklenen_no: byte;

implementation

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
begin
  quickrep1.Preview;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  close;
end;

procedure TForm1.Panel1DragDrop(Sender, Source:
  TObject; X, Y: Integer);
begin
  case suruklenen_no of
    1: begin
        label1.Top:=y;    qrlabel1.Top:=y;
        qrlabel1.left:=x;  label1.left:=x;
      end;
    2: begin
        label2.Top:=y;    qrlabel2.Top:=y;
        qrlabel2.left:=x;  label2.left:=x;
      end;
    3: begin
        label3.Top:=y;    qrlabel3.Top:=y;
        qrlabel3.left:=x;  label3.left:=x;
      end;
    4: begin
        label4.Top:=y;    qrlabel4.Top:=y;
        qrlabel4.left:=x;  label4.left:=x;
      end;
  end;
end;

procedure TForm1.Panel1DragOver(Sender, Source:
  TObject; X, Y: Integer;
    State: TDragState; var Accept: Boolean);
begin
  accept:=source is TLabel;
  case suruklenen_no of
    1: begin
        label1.Top:=y;    qrlabel1.Top:=y;
        qrlabel1.left:=x;  label1.left:=x;
      end;
    2: begin
        label2.Top:=y;    qrlabel2.Top:=y;
        qrlabel2.left:=x;  label2.left:=x;
      end;
    3: begin
        label3.Top:=y;    qrlabel3.Top:=y;
        qrlabel3.left:=x;  label3.left:=x;
      end;
  end;
end;

```

```

4: begin
    label4.Top:=y;    qrlabel4.Top:=y;
    qrlabel4.left:=x;    label4.left:=x;
end;
end;
end;

procedure TForm1.Button3Click(Sender: TObject); //
Güncelle düğmesi
begin
    qrlabel1.Font:=label1.Font;
    qrlabel1.Top:=label1.Top;
    qrlabel1.left:=label1.left;
    qrlabel1.Color:=label1.color;

    qrlabel2.Font:=label2.Font;
    qrlabel2.Top:=label2.Top;
    qrlabel2.left:=label2.left;
    qrlabel2.Color:=label2.color;

    qrlabel3.Font:=label3.Font;
    qrlabel3.Top:=label3.Top;
    qrlabel3.left:=label3.left;
    qrlabel3.Color:=label3.color;

    qrlabel4.Font:=label4.Font;
    qrlabel4.Top:=label4.Top;
    qrlabel4.left:=label4.left;
    qrlabel4.Color:=label4.color;
end;

procedure TForm1.Label3MouseDown(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin suruklenen_no:=3; end;

procedure TForm1.Label4MouseDown(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin suruklenen_no:=4; end;

procedure TForm1.Label1MouseDown(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin suruklenen_no:=1; end;

procedure TForm1.Label2MouseDown(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin suruklenen_no:=2; end;

procedure TForm1.Label1StartDrag(Sender: TObject;
var DragObject: TDragObject);
begin suruklenen_no:=1; end;

procedure TForm1.Label2StartDrag(Sender: TObject;
var DragObject: TDragObject);
begin suruklenen_no:=2; end;

```

```

procedure TForm1.Label3StartDrag(Sender: TObject;
var DragObject: TDragObject);
begin suruklenen_no:=3; end;

procedure TForm1.Label4StartDrag(Sender: TObject;
var DragObject: TDragObject);
begin suruklenen_no:=4; end;

procedure TForm1.YazTipi1Click(Sender: TObject);
begin
case suruklenen_no of
1: begin
    FontDialog1.Font:=label1.Font;
    FontDialog1.Execute;
    label1.Font:=FontDialog1.Font;
end;
2: begin
    FontDialog1.Font:=label2.Font;
    FontDialog1.Execute;
    label2.Font:=FontDialog1.Font;
end;
3: begin
    FontDialog1.Font:=label3.Font;
    FontDialog1.Execute;
    label3.Font:=FontDialog1.Font;
end;
4: begin
    FontDialog1.Font:=label4.Font;
    FontDialog1.Execute;
    label4.Font:=FontDialog1.Font;
end;
end;
end;

procedure TForm1.Renk1Click(Sender: TObject);
begin
case suruklenen_no of
1: begin
    ColorDialog1.Color:=label1.color;
    colorDialog1.Execute;    label1.color:=ColorDialog1.Color;
end;
2: begin
    ColorDialog1.Color:=label2.color;
    colorDialog1.Execute;    label2.color:=ColorDialog1.Color;
end;
3: begin
    ColorDialog1.Color:=label3.color;
    colorDialog1.Execute;    label3.color:=ColorDialog1.Color;
end;
4: begin
    ColorDialog1.Color:=label4.color;
    colorDialog1.Execute;    label4.color:=ColorDialog1.Color;
end;
end;
end;
end.

```

11. Thread (işlemciyi daha verimli kullanma)

(Ornekler\Delphi\Thread\Project1.dpr)

Windows'da dosya ararken vazgeçerseniz, "Dur" düğmesine basarsınız, *.wav dosyanızı *.mp3'e çevirirken de priority – işlem önceliği ile daha hızlı çevrim yapabilirsiniz. Bu tip işlemler Thread – kanallar ile yapılıyor. Yani siz uzun süren bir işleminizi kanala atarsınız, istediğiniz zaman duraksatır, devam ettirir veya iptal edebilirsiniz, hatta önemine göre Time Critical, High, Normal, Low, Idle yani en üst, üst, orta, alt ve en alt seviye işlemci kullanımı olarak ayarlayabilirsiniz.

İşletim sistemi o an çalışan process – işlemleri düzenlemekle görevlidir. Prosesler sistem prosesi ve kullanıcı prosesi olarak ikiye ayrılır. O anda çalışan uygulamalar sequential – sıralı (veya time sharing – zaman paylaşım) olarak işlenir. Birbirleri arasında belli anlarda önceliklerine göre geçiş yapılır. Yeni prosesler hazırda, beklemede veya çalışma hallerini geçerek sonlandırılmayı, yok edilmeyi beklerler. Fiziksel kaynak paylaşımı yani işlemci, RAM gibi donanım kullanılır, mantıksal kaynak paylaşımı ile ortak prosesler paylaşılır. Ana işlem, çocuk işlemler bitene dek bekler. Bir kanal ise işlemciyi kullanımının en basit birimidir. Kanallar işletim sistemi kaynaklarının, hafıza adreslerini ve program kodlarını paylaşırlar. Multitasking – çokgörevlilik sayesinde bir prosesin bitimini beklemek yerine, o proses idle – boş konuma geçtiği anda diğer proses uygulanır.

Normal bir application – uygulama projemize kanal özelliği vermek için "File*New...*New *Thread Object" diyoruz. Deneme ismindeki kanalımızı renk bilgisi yollayıp da çalıştırıyoruz. Yani parametrelili. Baslik ismindeki kanal ise bağımsız, ana programında ne olduğu onu ilgilendirmiyor.

Form başlığında devamlı değişen bir rakamla renkli noktalar basan thread'ın bilgisayarı ne kadar meşgul ettiğini görüyoruz. Combobox değiştirildikçe başlıktaki bu rakamların okunaklılığı değişiyor. Suspend – duraksama, Resume – devam etme ve formun kenarlarını boyutlandırınca da bu değişiklikleri görebilirsiniz. Tam ekran ve Highest – en üst öncelikte bayağı bilgisayarınız zorlanır.

Eğer programınız çok yoğun işlemlerle boğuşacak ve siz de bilgisayar o işle uğraşırken kilitlenmeden işlemlerini yapmasını isterseniz en emin yol bu multi thread – çoklu kanal kullanımıdır. Fakat gördüğünüz gibi karmaşık kod ve bilgisayarı sömürürcesine kullanması dezavantajlarıdır.

Farklı yöntemlerle de uzun sürecek işlemlerimizi yapabiliriz, bu yöntemlerle yapılan programda işlemi iptal etme gibi bir durum yoktur (Ornekler\Delphi\BACKPROC):

- 1) Fare imlecini hourglass – kumsaati haline getirerek, kullanıcıyı uyarmak. Tek yapacağınız işlem kodunun başına **Screen.Cursor:=crHourglass;** sonuna da **Screen.Cursor:=crDefault;** demek. Programa müdahale etmenin imkansız olduğunu unutmayın.

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
    Number, NPrimes, Max: LongInt;
```

```
begin
```

```
    Screen.Cursor := crHourglass;           //işte kahramanımız
```

```
    try
```

```
        NPrimes := 0;
```

```
        Max := SpinEdit1.Value;
```

```
        HourLabel.Caption := 'Hesaplanıyor...';
```

```
        for Number := 2 to Max do
```

```
            begin
```

```
                if IsPrime (Number) then           //IsPrime'in kodu üçüncü yöntemde
```

```
                    Inc (NPrimes);
```

```
                ProgressBar1.Position := Number * 100 div Max;
```

```
// aşağıdaki ikinci yöntem için bu satırı aktif yapabilirsiniz...
```

```
// Application.ProcessMessages;
```

```
    end;
```

```
    HourLabel.Caption := IntToStr (NPrimes) + ' below ' + IntToStr (Max);
```

```
finally
```

```
    Screen.Cursor := crDefault;           //normale dönüş
```

```
end; end;
```

- 2) Uygulamayı yavaşlatan ama hala müdahale imkanı kazandıran, uygun bir yere şu tek satırı eklemek: **Application.ProcessMessages**; Tüm işlemin bitmesini beklemeden tekrar çalıştırılabilir ki bu büyük sorunlara yol açabilir. Önlem olarak devam adında bir boolean değişken oluşturup tekrar çalıştırılmayı önleyebilirsiniz. Bu yöntem daha çok Win 3.1 uygulamalarında kullanılıyor.
- 3) Application nesnesinin OnIdle – atıl veya boş olduğu anları kollayıp, işlemleri yapan prosedür oluşturmak. Yavaşdır ama uygulamaya müdahale imkanı vardır:

```

...
procedure IdleButtonClick(Sender: TObject);
private
  IdleNumber, IdleNPrimes, IdleMax: LongInt;
public
  procedure IdleProc (Sender: TObject; var Done: Boolean);
  end;
var
  Form1: TForm1;
implementation

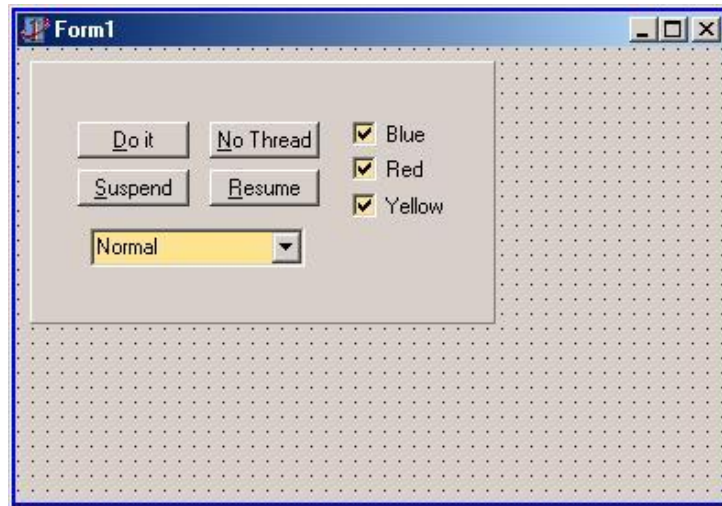
{$R *.DFM}

function IsPrime (N: LongInt): Boolean;
var
  Test: LongInt;
begin
  IsPrime := True;
  for Test := 2 to N - 1 do
  begin
    if (N mod Test) = 0 then
    begin
      IsPrime := False;
      break;
    end;
  end;
end;

procedure TForm1.IdleButtonClick(Sender: TObject);
begin
  IdleNPrimes := 0;
  IdleMax := SpinEdit1.Value;
  IdleNumber := 2;
  Application.OnIdle := IdleProc;           //atıl boş an çağrılır
  ProgressBar4.Position := 0;
  IdleLabel.Caption := 'Hesaplanıyor...';
end;

procedure TForm1.IdleProc (Sender: TObject; var Done: Boolean);
begin
  if IdleNumber < IdleMax then
  begin
    if IsPrime (IdleNumber) then
      Inc (IdleNPrimes);
    Inc (IdleNumber);
    Done := False;
  end else begin
    IdleLabel.Caption := IntToStr (IdleNPrimes) +
      ' below ' + IntToStr (IdleMax);
    Done := True;
    Beep;
    Application.OnIdle := nil;
  end;
  ProgressBar4.Position := IdleNumber * 100 div IdleMax;
end;
end.

```

Resim 19 – Thread formumuz

Proje dosyamız:

```
program Project1;

uses
  Forms,
  Unit1 in 'Unit1.pas' {Form1},
  Unit2 in 'Unit2.pas',
  Unit3 in 'Unit3.pas';

{$R *.RES}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
```

Unit1.pas yani ana programımız:

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, Unit2, Unit3;

type
  TForm1 = class(TForm)
    Panel1: TPanel;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    ComboBox1: TComboBox;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    CheckBox3: TCheckBox;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
  private
```

```

    { Private declarations }
public
    { Public declarations }
    BenimThread:array [1..3] of TDeneme;
    bslk:Baslik;
end;

var
    Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
begin
    Button1.Enabled := False;
    Button2.Enabled := True;
    Button3.Enabled := true;
    Button4.Enabled := true;
    ComboBox1.Enabled := true;
    CheckBox1.Enabled := false;
    CheckBox2.Enabled := false;
    CheckBox3.Enabled := false;
    if CheckBox1.Checked then BenimThread[1]:=Tdeneme.Create(clblue);
    if CheckBox2.Checked then BenimThread[2]:=Tdeneme.Create(clred);
    if CheckBox3.Checked then BenimThread[3]:=Tdeneme.Create(clyellow);
    bslk:=baslik.Create(False);
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
    if CheckBox1.Checked then BenimThread[1].Free;
    if CheckBox2.Checked then BenimThread[2].Free;
    if CheckBox3.Checked then BenimThread[3].Free;
    Button1.Enabled := True;
    Button2.Enabled := False;
    Button3.Enabled := false;
    Button4.Enabled := false;
    ComboBox1.Enabled := false;
    CheckBox1.Enabled := true;
    CheckBox2.Enabled := true;
    CheckBox3.Enabled := true;
    bslk.free;
    Caption:='Form1';
    form1.Refresh;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
    if CheckBox1.Checked then BenimThread[1].suspend;
    if CheckBox2.Checked then BenimThread[2].suspend;
    if CheckBox3.Checked then BenimThread[3].suspend;
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
    if CheckBox1.Checked then BenimThread[1].resume;
    if CheckBox2.Checked then BenimThread[2].resume;
    if CheckBox3.Checked then BenimThread[3].resume;
end;

procedure TForm1.ComboBox1Change(Sender: TObject);
begin
    if CheckBox1.Checked then BenimThread[1].Suspend;
    if CheckBox2.Checked then BenimThread[2].Suspend;

```

```

    if CheckBox3.Checked then BenimThread[3].Suspend;
if ComboBox1.ItemIndex=0 then begin
    if CheckBox1.Checked then BenimThread[1].Priority:=tpTimeCritical;
    if CheckBox2.Checked then BenimThread[2].Priority:=tpTimeCritical;
    if CheckBox3.Checked then BenimThread[3].Priority:=tpTimeCritical;
end;
if ComboBox1.ItemIndex=1 then begin
    if CheckBox1.Checked then BenimThread[1].Priority:=tpHighest;
    if CheckBox2.Checked then BenimThread[2].Priority:=tpHighest;
    if CheckBox3.Checked then BenimThread[3].Priority:=tpHighest;
end;
if ComboBox1.ItemIndex=2 then begin
    if CheckBox1.Checked then BenimThread[1].Priority:=tpHigher;
    if CheckBox2.Checked then BenimThread[2].Priority:=tpHigher;
    if CheckBox3.Checked then BenimThread[3].Priority:=tpHigher;
end;
if ComboBox1.ItemIndex=3 then begin
    if CheckBox1.Checked then BenimThread[1].Priority:=tpNormal;
    if CheckBox2.Checked then BenimThread[2].Priority:=tpNormal;
    if CheckBox3.Checked then BenimThread[3].Priority:=tpNormal;
end;
if ComboBox1.ItemIndex=4 then begin
    if CheckBox1.Checked then BenimThread[1].Priority:=tpLower;
    if CheckBox2.Checked then BenimThread[2].Priority:=tpLower;
    if CheckBox3.Checked then BenimThread[3].Priority:=tpLower;
end;
if ComboBox1.ItemIndex=5 then begin
    if CheckBox1.Checked then BenimThread[1].Priority:=tpLowest;
    if CheckBox2.Checked then BenimThread[2].Priority:=tpLowest;
    if CheckBox3.Checked then BenimThread[3].Priority:=tpLowest;
end;
if ComboBox1.ItemIndex=6 then begin
    if CheckBox1.Checked then BenimThread[1].Priority:=tpIdle;
    if CheckBox2.Checked then BenimThread[2].Priority:=tpIdle;
    if CheckBox3.Checked then BenimThread[3].Priority:=tpIdle;
end;
    if CheckBox1.Checked then BenimThread[1].resume;
    if CheckBox2.Checked then BenimThread[2].resume;
    if CheckBox3.Checked then BenimThread[3].resume;
end;
end.

```

Unit2.pas renkli nokta basan thread kodlarımız:

```

unit Unit2;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    StdCtrls;

type
    TDeneme = class(TThread)
    private
        { Private declarations }
        renk:integer;
    protected
        procedure Execute; override;
    public
        constructor Create(color:Tcolor);
    end;

implementation

{ TDeneme }

```

```

uses unit1;

constructor Tdeneme.Create(color:Tcolor);
begin
    renk:=color;
    inherited Create(false);
end;

procedure TDeneme.Execute;
var
    X, Y: Integer;
begin
    Randomize;
    repeat
        X := Random (Form1.ClientWidth);
        Y := Random (Form1.ClientHeight);
        with Form1.Canvas do
            begin
                Lock;
                try
                    Pixels [X, Y] := renk;
                finally
                    Unlock;
                end;
            end;
        until Terminated;
    end;
end.

```

Unit3.pas formumuzun başlığına bilgi yazan thread:

```

unit Unit3;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    StdCtrls,Unit2, ExtCtrls;

type
    Baslik = class(TThread)
    protected
        procedure Execute; override;
    end;

implementation

{ Baslik }
uses unit1;

var i:integer;

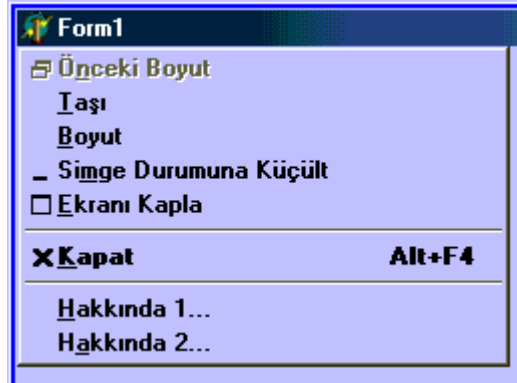
procedure Baslik.Execute;
begin
    while not Terminated do begin
        for i:=1 to 100 do
            form1.caption:='Form1 i='+inttostr(i);
        end;
    end;
end.

```

12. Sysmenu (sistem menüsüne eleman eklenmesi)

(Ornekler\Delphi\sysmenu\Project1.dpr)

Sistem menüsüne kendi komutlarınızı eklemek için şu örneği yapabilirsiniz. Fakat sağ tarafta aktif hale getirebildiğimiz "Bu Nedir?" düğmesinin, Yardım dosyanız ile nasıl ilişkilendirileceğinden burada bahsetmeyeceğim. Bu ek Başlat çubuğundaki programın simgesindeki sistem menüsünü etkilemiyor.



Resim 20 – Sistem menüsü

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;

type
  TForm1 = class(TForm)
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    procedure WMSysCommand (var msg:TWMSysCommand);
    message wm_SysCommand;
  end;

var
  Form1: TForm1;

implementation

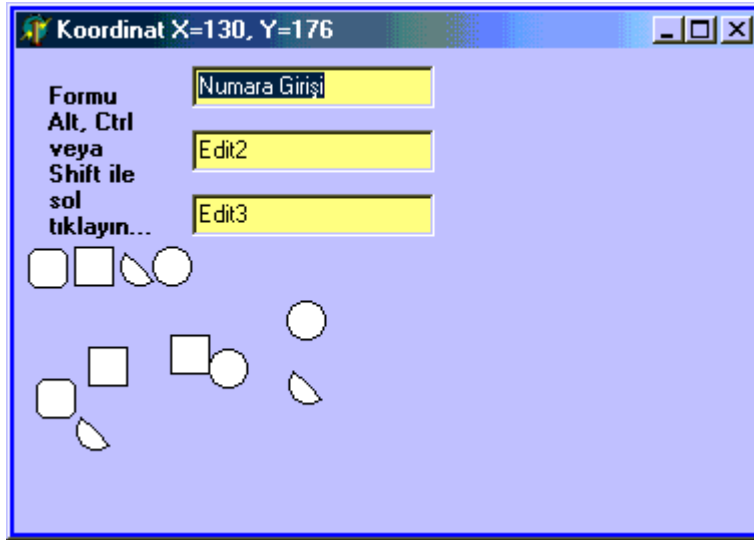
{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
  AppendMenu (GetSystemMenu (Handle, FALSE), MF_SEPARATOR, 0, "");
  AppendMenu (GetSystemMenu (Handle, FALSE), MF_STRING, 1, '&Hakkında 1...');
  AppendMenu (GetSystemMenu (Handle, FALSE), MF_STRING, 2, 'H&akkında 2...');
end;

procedure TForm1.WMSysCommand (var Msg: TWMSysCommand);
begin
  if Msg.CmdType = 1 then ShowMessage ('Meraba!'); if Msg.CmdType = 2 then ShowMessage ('Selam!');
  inherited;
end;
end.
```

13. Tuş Basımı (klavye ve fare basımlarını kontrol etme)

(Ornekler\Delphi\tusbasimi\Project1.dpr)



Resim 21 – Fare ve klavye basımı örneği

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;

type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Label1: TLabel;
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
    procedure FormMouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Edit2KeyPress(Sender: TObject; var Key: Char);
    procedure Edit3KeyPress(Sender: TObject; var Key: Char);
    procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure FormDbClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
  if key=#13 then begin
    key:=#0; Perform (CM_DialogKey, VK_TAB,0);    //tab efekti
```

```

end;
if not (key in ['0'..'9',#8]) then begin      //sadece rakam girişı
    key:=#0; beep;
end;
end;

```

```

procedure TForm1.FormMouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    if button = mleft then
        if (ssCtrl in Shift) and (ssShift in shift) then
            canvas.RoundRect(x-10, y-10, x+10, y+10 ,10, 10 )
        else
            if ssShift in Shift then
                canvas.Rectangle (x-10, y-10, x+10, y+10)
            else
                if ssAlt in Shift then
                    canvas.Ellipse (x-10, y-10, x+10, y+10)
                else
                    if ssCtrl in Shift then
                        canvas.Pie (x-10, y-10, x+10, y+10 ,x-10, y-10, x+10, y+10 );
                    end;
                end;
            end;
        end;
    end;

```

```

procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
    if key=#13 then begin
        key:=#0; Perform (CM_DialogKey, VK_TAB,0);
    end;
end;

```

```

procedure TForm1.Edit3KeyPress(Sender: TObject; var Key: Char);
begin
    if key=#13 then begin
        key:=#0; Perform (CM_DialogKey, VK_TAB,0);
    end;
end;

```

```

procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
begin
    caption:=format('Koordinat X=%3d, Y=%3d',[x,y]);
end;

```

```

procedure TForm1.FormDbIClick(Sender: TObject);
begin
    repaint;          // hem invalidate hem de update yapılır (=refresh)
end;

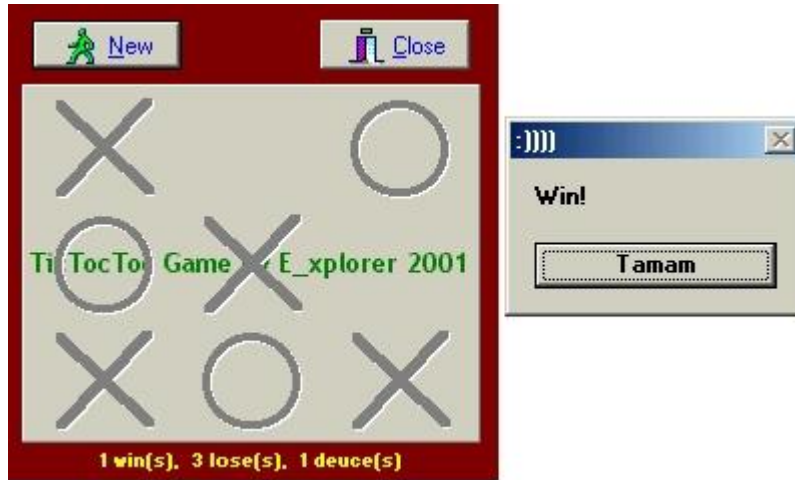
```

end.

Form üzerinde bir adet açıklama Label'ı ve üç adet de Edit nesnesi vardır. Yine form üzerinde fare ve klavye tuşları basımı ile şekiller çıkartılıyor. Editlerde ise Enter tuşu basımı Tab efekti kazanıyor. MS Paint örneği fare ile şekil çizme için daha kullanışlı.

14. 2 Boyutlu Oyun Yapımı (TicTacToe, Pazl, Mayn oyunları)

(Ornekler\Delphi\tictoc\Project1.dpr)



Resim 22a – TicTacToe Oyunumuzun ekran görüntüsü

```
object Form1: TForm1
  Left = 365
  Top = 110
  BorderIcons = [biSystemMenu, biMinimize]
  BorderStyle = bsNone
  ClientHeight = 238
  ClientWidth = 246
  Color = clMaroon
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  OldCreateOrder = False
  Position = poDesktopCenter
  OnCreate = FormCreate
  OnMouseDown = FormMouseDown
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
    Left = 9
    Top = 222
    Width = 224
    Height = 11
    Alignment = taCenter
    AutoSize = False
    Caption = 'Tic Tac Toe Game by E_xplorer 2001'
    Font.Charset = TURKISH_CHARSET
    Font.Color = clYellow
    Font.Height = -9
    Font.Name = 'Small Fonts'
    Font.Style = [fsBold]
    ParentFont = False
    Transparent = True
  end
  object BitBtn1: TBitBtn
    Left = 157
    Top = 8
    Width = 75
    Height = 25
    Font.Charset = DEFAULT_CHARSET
```

```
    Font.Color = clBlue
    Font.Height = -11
    Font.Name = 'MS Sans Serif'
    Font.Style = []
    ParentFont = False
    TabOrder = 0
    Kind = bkClose
  end
  object BitBtn2: TBitBtn
    Left = 13
    Top = 8
    Width = 75
    Height = 25
    Caption = '&New'
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clBlue
    Font.Height = -11
    Font.Name = 'MS Sans Serif'
    Font.Style = []
    ParentFont = False
    TabOrder = 1
    OnClick = BitBtn2Click
    Kind = bkIgnore
  end
  object Panel1: TPanel
    Left = 8
    Top = 40
    Width = 229
    Height = 179
    Caption = 'Ciko'#39's TicTocToe Game by E_xplorer 2001'
    Font.Charset = TURKISH_CHARSET
    Font.Color = clGreen
    Font.Height = -13
    Font.Name = 'Trebuchet MS'
    Font.Style = [fsBold]
    ParentFont = False
    TabOrder = 2
  end
  object SpeedButton7: TSpeedButton
    Left = 4
    Top = 119
```



```

Width = 73
Height = 57
Flat = True
OnClick = SpeedButton7Click
end
object SpeedButton8: TSpeedButton
Left = 78
Top = 119
Width = 73
Height = 57
Flat = True
OnClick = SpeedButton8Click
end
object SpeedButton5: TSpeedButton
Left = 78
Top = 61
Width = 73
Height = 57
Flat = True
OnClick = SpeedButton5Click
end
object SpeedButton6: TSpeedButton
Left = 152
Top = 61
Width = 73
Height = 57
Flat = True
OnClick = SpeedButton6Click
end
object SpeedButton9: TSpeedButton
Left = 152
Top = 119
Width = 73
Height = 57
Flat = True
OnClick = SpeedButton9Click

```

```

end
object SpeedButton4: TSpeedButton
Left = 4
Top = 61
Width = 73
Height = 57
Flat = True
OnClick = SpeedButton4Click
end
object SpeedButton3: TSpeedButton
Left = 152
Top = 3
Width = 73
Height = 57
Flat = True
OnClick = SpeedButton3Click
end
object SpeedButton2: TSpeedButton
Left = 78
Top = 3
Width = 73
Height = 57
Flat = True
OnClick = SpeedButton2Click
end
object SpeedButton1: TSpeedButton
Left = 4
Top = 3
Width = 73
Height = 57
Flat = True
OnClick = SpeedButton1Click
end
end
end

```

Kodlarımız ise:

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
Buttons, StdCtrls, ExtCtrls;
```

```
type
```

```

TForm1 = class(TForm)
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
Label1: TLabel;
Panel1: TPanel;
SpeedButton1: TSpeedButton;
SpeedButton2: TSpeedButton;
SpeedButton3: TSpeedButton;
SpeedButton4: TSpeedButton;
SpeedButton5: TSpeedButton;
SpeedButton6: TSpeedButton;
SpeedButton7: TSpeedButton;
SpeedButton8: TSpeedButton;
SpeedButton9: TSpeedButton;
procedure BitBtn2Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FormMouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);

```

```

procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
procedure SpeedButton4Click(Sender: TObject);
procedure SpeedButton5Click(Sender: TObject);
procedure SpeedButton6Click(Sender: TObject);
procedure SpeedButton7Click(Sender: TObject);
procedure SpeedButton8Click(Sender: TObject);
procedure SpeedButton9Click(Sender: TObject);
procedure kontrol;
procedure disable;
procedure toplam;
function topcheck: byte;
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  basili: array [1..9] of char;
  sirakimde:boolean;
  toplamtop:byte;
  wins,loses,deuces:byte;
implementation

{$R *.DFM}
procedure tform1.toplam;
var i : byte;
begin
  toplamtop:=0;
  for i:= 1 to 9 do begin
    if basili[i] <> '-' then inc(toplamtop);
  end;
end;

Function tform1.topcheck() : byte;
var a:byte;
begin
  while true do begin
    a:= random(9);
    topcheck:=a;                      //recursive...
    if basili[a]='-' then exit;
  end;
end;

procedure Tform1.disable;
begin
  SpeedButton1.Enabled:=false;
  SpeedButton2.Enabled:=false;
  SpeedButton3.Enabled:=false;
  SpeedButton4.Enabled:=false;
  SpeedButton5.Enabled:=false;
  SpeedButton6.Enabled:=false;
  SpeedButton7.Enabled:=false;
  SpeedButton8.Enabled:=false;
  SpeedButton9.Enabled:=false;
end;

procedure Tform1.kontrol;
var pc:byte;
begin
if (basili[1]='x') and (basili[2]='x') and (basili[3]='x') then
  begin
    inc(wins);
  end;
end;

```

```

label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox("Win!',':::))',0);
disable;exit;
end;
if (basili[1]='x') and (basili[4]='x') and (basili[7]='x') then
begin
inc(wins);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox("Win!',':::))',0);
disable;exit;
end;
if (basili[7]='x') and (basili[8]='x') and (basili[9]='x') then
begin
inc(wins);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox("Win!',':::))',0);
disable;exit;
end;
if (basili[3]='x') and (basili[6]='x') and (basili[9]='x') then
begin
inc(wins);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox("Win!',':::))',0);
disable;exit;
end;
if (basili[1]='x') and (basili[5]='x') and (basili[9]='x') then
begin
inc(wins);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox("Win!',':::))',0);
disable;exit;
end;
if (basili[3]='x') and (basili[5]='x') and (basili[7]='x') then
begin
inc(wins);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox("Win!',':::))',0);
disable;exit;
end;
if (basili[2]='x') and (basili[5]='x') and (basili[8]='x') then
begin
inc(wins);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox("Win!',':::))',0);
disable;exit;
end;
if (basili[4]='x') and (basili[5]='x') and (basili[6]='x') then
begin
inc(wins);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox("Win!',':::))',0);
disable;exit;
end;

toplamlam;
if toplamtop = 9 then
begin
inc(deuces);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox("Deuce!','$±$±$',0);
disable;exit;
end;

pc:=0; //hiç top seçili değil
if toplamtop = 1 then pc:= topcheck() //daha oyunun başındayız
else begin

```

```

if (basili[1]='x') and (basili[2]='x') and (basili[3]='-') then pc:=3;
if (basili[2]='x') and (basili[3]='x') and (basili[1]='-') then pc:=1;
if (basili[4]='x') and (basili[5]='x') and (basili[6]='-') then pc:=6;
if (basili[5]='x') and (basili[6]='x') and (basili[4]='-') then pc:=4;
if (basili[7]='x') and (basili[8]='x') and (basili[9]='-') then pc:=9;
if (basili[8]='x') and (basili[9]='x') and (basili[7]='-') then pc:=7;
if (basili[1]='x') and (basili[4]='x') and (basili[7]='-') then pc:=7;
if (basili[4]='x') and (basili[7]='x') and (basili[1]='-') then pc:=1;
if (basili[2]='x') and (basili[5]='x') and (basili[8]='-') then pc:=8;
if (basili[5]='x') and (basili[8]='x') and (basili[2]='-') then pc:=2;
if (basili[3]='x') and (basili[6]='x') and (basili[9]='-') then pc:=9;
if (basili[6]='x') and (basili[9]='x') and (basili[3]='-') then pc:=3;

```

```

if (basili[1]='x') and (basili[5]='x') and (basili[9]='-') then pc:=9;
if (basili[5]='x') and (basili[9]='x') and (basili[1]='-') then pc:=1;
if (basili[3]='x') and (basili[5]='x') and (basili[7]='-') then pc:=7;
if (basili[7]='x') and (basili[5]='x') and (basili[3]='-') then pc:=3;

```

```

if (basili[1]='x') and (basili[3]='x') and (basili[2]='-') then pc:=2;
if (basili[1]='x') and (basili[7]='x') and (basili[4]='-') then pc:=4;
if (basili[3]='x') and (basili[9]='x') and (basili[6]='-') then pc:=6;
if (basili[7]='x') and (basili[9]='x') and (basili[8]='-') then pc:=8;

```

```

if (basili[4]='x') and (basili[6]='x') and (basili[5]='-') then pc:=5;
if (basili[2]='x') and (basili[8]='x') and (basili[5]='-') then pc:=5;

```

```

if (basili[1]='x') and (basili[9]='x') and (basili[5]='-') then pc:=5;
if (basili[3]='x') and (basili[7]='x') and (basili[5]='-') then pc:=5;

```

```
//---
```

```

if (basili[1]='o') and (basili[2]='o') and (basili[3]='-') then pc:=3;
if (basili[2]='o') and (basili[3]='o') and (basili[1]='-') then pc:=1;
if (basili[4]='o') and (basili[5]='o') and (basili[6]='-') then pc:=6;
if (basili[5]='o') and (basili[6]='o') and (basili[4]='-') then pc:=4;
if (basili[7]='o') and (basili[8]='o') and (basili[9]='-') then pc:=9;
if (basili[8]='o') and (basili[9]='o') and (basili[7]='-') then pc:=7;

```

```

if (basili[1]='o') and (basili[4]='o') and (basili[7]='-') then pc:=7;
if (basili[4]='o') and (basili[7]='o') and (basili[1]='-') then pc:=1;
if (basili[2]='o') and (basili[5]='o') and (basili[8]='-') then pc:=8;
if (basili[5]='o') and (basili[8]='o') and (basili[2]='-') then pc:=2;
if (basili[3]='o') and (basili[6]='o') and (basili[9]='-') then pc:=9;
if (basili[6]='o') and (basili[9]='o') and (basili[3]='-') then pc:=3;

```

```

if (basili[1]='o') and (basili[5]='o') and (basili[9]='-') then pc:=9;
if (basili[5]='o') and (basili[9]='o') and (basili[1]='-') then pc:=1;
if (basili[3]='o') and (basili[5]='o') and (basili[7]='-') then pc:=7;
if (basili[7]='o') and (basili[5]='o') and (basili[3]='-') then pc:=3;

```

```

if (basili[1]='o') and (basili[3]='o') and (basili[2]='-') then pc:=2;
if (basili[1]='o') and (basili[7]='o') and (basili[4]='-') then pc:=4;
if (basili[3]='o') and (basili[9]='o') and (basili[6]='-') then pc:=6;
if (basili[7]='o') and (basili[9]='o') and (basili[8]='-') then pc:=8;

```

```

if (basili[4]='o') and (basili[6]='o') and (basili[5]='-') then pc:=5;
if (basili[2]='o') and (basili[8]='o') and (basili[5]='-') then pc:=5;

```

```

if (basili[1]='o') and (basili[9]='o') and (basili[5]='-') then pc:=5;
if (basili[3]='o') and (basili[7]='o') and (basili[5]='-') then pc:=5;
end; //üst kısımda o harfi kullanıyoruz, sıfır değil...

```

```
if pc = 0 then pc:=topcheck();
```

```
if pc = 1 then
begin
```

```
    basili[1]:='o';
    beep;
```

```

        SpeedButton1.Enabled:=false;
        SpeedButton1.Glyph.loadfromfile('o.bmp');
    end;
    if pc = 2 then
    begin
        basili[2]:='o';
        beep;
        SpeedButton2.Enabled:=false;
        SpeedButton2.Glyph.loadfromfile('o.bmp');
    end;
    if pc = 3 then
    begin
        basili[3]:='o';
        beep;
        SpeedButton3.Enabled:=false;
        SpeedButton3.Glyph.loadfromfile('o.bmp');
    end;
    if pc = 4 then
    begin
        basili[4]:='o';
        beep;
        SpeedButton4.Enabled:=false;
        SpeedButton4.Glyph.loadfromfile('o.bmp');
    end;
    if pc = 5 then
    begin
        basili[5]:='o';
        beep;
        SpeedButton5.Enabled:=false;
        SpeedButton5.Glyph.loadfromfile('o.bmp');
    end;
    if pc = 6 then
    begin
        basili[6]:='o';
        beep;
        SpeedButton6.Enabled:=false;
        SpeedButton6.Glyph.loadfromfile('o.bmp');
    end;
    if pc = 7 then
    begin
        basili[7]:='o';
        beep;
        SpeedButton7.Enabled:=false;
        SpeedButton7.Glyph.loadfromfile('o.bmp');
    end;
    if pc = 8 then
    begin
        basili[8]:='o';
        beep;
        SpeedButton8.Enabled:=false;
        SpeedButton8.Glyph.loadfromfile('o.bmp');
    end;
    if pc = 9 then
    begin
        basili[9]:='o';
        beep;
        SpeedButton9.Enabled:=false;
        SpeedButton9.Glyph.loadfromfile('o.bmp');
    end;
    if (basili[1]='o') and (basili[2]='o') and (basili[3]='o') then
    begin
        inc(loses);
        label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
        Application.MessageBox('Lose!',':((((',0);
        disable;exit;
    end;
end;

```

```

if (basili[1]='o') and (basili[4]='o') and (basili[7]='o') then
begin
inc(loses);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox('Lose!',':(((((' ,0);
disable;exit;
end;
if (basili[7]='o') and (basili[8]='o') and (basili[9]='o') then
begin
inc(loses);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox('Lose!',':(((((' ,0);
disable;exit;
end;
if (basili[3]='o') and (basili[6]='o') and (basili[9]='o') then
begin
inc(loses);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox('Lose!',':(((((' ,0);
disable;exit;
end;
if (basili[1]='o') and (basili[5]='o') and (basili[9]='o') then
begin
inc(loses);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox('Lose!',':(((((' ,0);
disable;exit;
end;
if (basili[3]='o') and (basili[5]='o') and (basili[7]='o') then
begin
inc(loses);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox('Lose!',':(((((' ,0);
disable;exit;
end;
if (basili[2]='o') and (basili[5]='o') and (basili[8]='o') then
begin
inc(loses);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox('Lose!',':(((((' ,0);
disable;exit;
end;
if (basili[4]='o') and (basili[5]='o') and (basili[6]='o') then
begin
inc(loses);
label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[wins,loses,deuces]);
Application.MessageBox('Lose!',':(((((' ,0);
disable;exit;
end;
end;

procedure TForm1.BitBtn2Click(Sender: TObject);
var i:byte;
begin
for i:= 1 to 9 do basili[i]:='-';
SpeedButton1.Glyph.LoadFromFile('bos.bmp');
SpeedButton2.Glyph.LoadFromFile('bos.bmp');
SpeedButton3.Glyph.LoadFromFile('bos.bmp');
SpeedButton4.Glyph.LoadFromFile('bos.bmp');
SpeedButton5.Glyph.LoadFromFile('bos.bmp');
SpeedButton6.Glyph.LoadFromFile('bos.bmp');
SpeedButton7.Glyph.LoadFromFile('bos.bmp');
SpeedButton8.Glyph.LoadFromFile('bos.bmp');
SpeedButton9.Glyph.LoadFromFile('bos.bmp');
SpeedButton1.Enabled:=true;
SpeedButton2.Enabled:=true;

```

```

SpeedButton3.Enabled:=true;
SpeedButton4.Enabled:=true;
SpeedButton5.Enabled:=true;
SpeedButton6.Enabled:=true;
SpeedButton7.Enabled:=true;
SpeedButton8.Enabled:=true;
SpeedButton9.Enabled:=true;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  Randomize;
  BitBtn2.Click;
  sirakimde:=true;
  label1.Caption:=format('%2d win(s), %2d lose(s), %2d deuce(s)',[ wins , loses , deuces ] );
  panel1.Caption:= 'TicTocToe Game by E_xplorer 2001';
end;

procedure TForm1.FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
const
  SC_DragMove = $F012;
begin
  ReleaseCapture;
  FORM1.perform(WM_SysCommand, SC_DragMove, 0);
end;

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
  basili[1]:='x';
  beep;
  SpeedButton1.Enabled:=false;
  SpeedButton1.Glyph.loadfromfile('x.bmp');
  sirakimde:=false;
  kontrol;
end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
  basili[2]:='x';
  beep;
  SpeedButton2.Enabled:=false;
  SpeedButton2.Glyph.loadfromfile('x.bmp');
  sirakimde:=false;
  kontrol;
end;

procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
  basili[3]:='x';
  beep;
  SpeedButton3.Enabled:=false;
  SpeedButton3.Glyph.loadfromfile('x.bmp');
  sirakimde:=false;
  kontrol;
end;

procedure TForm1.SpeedButton4Click(Sender: TObject);
begin
  basili[4]:='x';
  beep;
  SpeedButton4.Enabled:=false;
  SpeedButton4.Glyph.loadfromfile('x.bmp');
  sirakimde:=false;
  kontrol;
end;

```

```

procedure TForm1.SpeedButton5Click(Sender: TObject);
begin
  basili[5]:='x';
  beep;
  SpeedButton5.Enabled:=false;
  SpeedButton5.Glyph.loadfromfile('x.bmp');
  sirakimde:=false;
  kontrol;
end;

```

```

procedure TForm1.SpeedButton6Click(Sender: TObject);
begin
  basili[6]:='x';
  beep;
  SpeedButton6.Enabled:=false;
  SpeedButton6.Glyph.loadfromfile('x.bmp');
  sirakimde:=false;
  kontrol;
end;

```

```

procedure TForm1.SpeedButton7Click(Sender: TObject);
begin
  basili[7]:='x';
  beep;
  SpeedButton7.Enabled:=false;
  SpeedButton7.Glyph.loadfromfile('x.bmp');
  sirakimde:=false;
  kontrol;
end;

```

```

procedure TForm1.SpeedButton8Click(Sender: TObject);
begin
  basili[8]:='x';
  beep;
  SpeedButton8.Enabled:=false;
  SpeedButton8.Glyph.loadfromfile('x.bmp');
  sirakimde:=false;
  kontrol;
end;

```

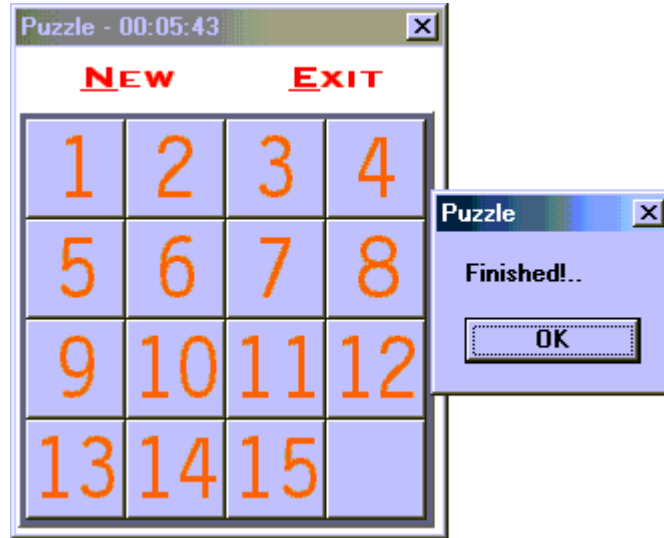
```

procedure TForm1.SpeedButton9Click(Sender: TObject);
begin
  basili[9]:='x';
  beep;
  SpeedButton9.Enabled:=false;
  SpeedButton9.Glyph.loadfromfile('x.bmp');
  sirakimde:=false;
  kontrol;
end;
end.

```

50x50 ölçeğinde x.bmp, o.bmp ve bos.bmp oluşturuyoruz. Bunları isterseniz Delphi'nin yardımcı programlarından olan Image Editor ile rahatlıkla yapabilirsiniz. Webdings fontundaki **×** (r harfi) ve **Θ** (y harfi) simgeleri de resim gibi kullanılabilir. Flat – düz özelliği nedeni ile 3x3 dizi halinde dokuz adet SpeedButton kullanıyoruz. Asıl oyunda kullanıcı ve bilgisayarın şekilleri her oyunda yer değiştiriyor, fakat fazla uğraşmadan kullanıcıya "x", bilgisayara da "o" işaretini ayırdım. Kod kısmı yine çok uzun gelebilir. Bilgisayara iyi düşünmeyi öğretmek zordur, fakat öğrenince de o asla hata yapmaz. Eminim siz daha kısa ve optimum kod üretebilirsiniz.

Yine benzer bir mantık ile Puzzle isimli bir bulmaca oyunu yapabilirsiniz. Delphi 5 dili kullanarak yaptığım projeyi CD'de "Ornekler\Delphi\Pazl\Project2.dpr" adı ile bulabilirsiniz. Temel mantık boş hücreyi kullanarak, rasgele dizilmiş hücreleri sıralamak. Program kısmı yine çok uzun, fakat aslında sadece SpeedButton nesnelerinin Caption özellikleri ile oynayıp sonuca varıyoruz. Hangisi tıklanmış ise etrafında boş hücre var mı şeklinde kontrol yapısı var. İsterseniz rakamları dizmek yerine bir resmi birleştirmek daha zevkli ve daha zorlayıcı olabilir. "Ornekler\Delphi\Pazl\resim\Project2.dpr" bunun resimli hali diyebiliriz.



Resim 22b – Puzzle Oyunumuzun ekran görüntüsü

Form üzerinde 4x4 yani 16 adet SpeedButton oyun hücresi için, 2 adet SpeedButton menü için ve bir adet de Timer1 nesnesi var. Timer1 ile kazanma anı ve süre kontrol ediliyor.

Kodlarımız:

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, ExtCtrls, Buttons, StdCtrls;
```

```
type
  TForm1 = class(TForm)
    Panel1: TPanel;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    SpeedButton5: TSpeedButton;
    SpeedButton6: TSpeedButton;
    SpeedButton7: TSpeedButton;
    SpeedButton8: TSpeedButton;
    SpeedButton9: TSpeedButton;
    SpeedButton10: TSpeedButton;
    SpeedButton11: TSpeedButton;
    SpeedButton12: TSpeedButton;
    SpeedButton13: TSpeedButton;
    SpeedButton14: TSpeedButton;
    SpeedButton15: TSpeedButton;
    SpeedButton16: TSpeedButton;
    SpeedButton17: TSpeedButton;
    SpeedButton18: TSpeedButton;
    Timer1: TTimer;
    procedure SpeedButton16Click(Sender: TObject);
    procedure SpeedButton17Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure SpeedButton15Click(Sender: TObject);
    procedure SpeedButton18Click(Sender: TObject);
    procedure SpeedButton14Click(Sender: TObject);
    procedure SpeedButton13Click(Sender: TObject);
    procedure SpeedButton9Click(Sender: TObject);
    procedure SpeedButton10Click(Sender: TObject);
    procedure SpeedButton11Click(Sender: TObject);
```

```
    procedure SpeedButton12Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure SpeedButton6Click(Sender: TObject);
    procedure SpeedButton7Click(Sender: TObject);
    procedure SpeedButton8Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
type //kendi değişken tipimizi oluşturuyoruz
  hücre=record
    deger:integer;
    bos:boolean;
  end;
var
  Form1: TForm1;
  dizi:array [1..16] of hücre; //16 adet hücre var
  sayac:byte;
  zaman:TTime;
implementation

{$R *.DFM}
procedure TForm1.SpeedButton16Click(Sender: TObject);
begin close end;

procedure tusdoldur;
//rakamların sadece bir kere çıkabilmesi için
var a:integer;
begin
  repeat
    a:=random(15)+1;
    if dizi[a].bos then
```

```

begin
    dizi[a].bos:=false;
    dizi[a].deger:=sayac;
    inc(sayac);
end
else tusdoldur;
until sayac=16;
end;

procedure TForm1.SpeedButton17Click(Sender: TObject);
var i:integer;
begin
    sayac:=1;
    dizi[1].bos:=true;dizi[2].bos:=true;dizi[3].bos:=true;
    dizi[4].bos:=true;dizi[5].bos:=true;dizi[6].bos:=true;
    dizi[7].bos:=true;dizi[8].bos:=true;dizi[9].bos:=true;
    dizi[10].bos:=true;dizi[11].bos:=true;dizi[12].bos:=true;
    dizi[13].bos:=true;dizi[14].bos:=true;dizi[15].bos:=true;
    dizi[16].bos:=true;
    tusdoldur;
    SpeedButton1.Caption:=inttostr(dizi[1].deger);
    SpeedButton2.Caption:=inttostr(dizi[2].deger);
    SpeedButton3.Caption:=inttostr(dizi[3].deger);
    SpeedButton4.Caption:=inttostr(dizi[4].deger);
    SpeedButton5.Caption:=inttostr(dizi[5].deger);
    SpeedButton6.Caption:=inttostr(dizi[6].deger);
    SpeedButton7.Caption:=inttostr(dizi[7].deger);
    SpeedButton8.Caption:=inttostr(dizi[8].deger);
    SpeedButton9.Caption:=inttostr(dizi[9].deger);
    SpeedButton10.Caption:=inttostr(dizi[10].deger);
    SpeedButton11.Caption:=inttostr(dizi[11].deger);
    SpeedButton12.Caption:=inttostr(dizi[12].deger);
    SpeedButton13.Caption:=inttostr(dizi[13].deger);
    SpeedButton14.Caption:=inttostr(dizi[14].deger);
    SpeedButton15.Caption:=inttostr(dizi[15].deger);
    SpeedButton18.Caption:=' ';
    timer1.Enabled:=true;
    zaman:=now();
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    Randomize;Application.HintColor:=clyellow;
end;

procedure TForm1.SpeedButton15Click(Sender: TObject);
begin
    if SpeedButton18.Caption=' ' then
        //iki tek tırnak aralarını hep bir boşluk ile doldurun...
        begin
            SpeedButton18.Caption:=SpeedButton15.Caption;
            SpeedButton15.Caption:=' ';
        end else
        if SpeedButton11.Caption=' ' then
            begin
                SpeedButton11.Caption:=SpeedButton15.Caption;
                SpeedButton15.Caption:=' ';
            end else
        if SpeedButton14.Caption=' ' then
            begin
                SpeedButton14.Caption:=SpeedButton15.Caption;
                SpeedButton15.Caption:=' ';
            end;
        end;
end;

procedure TForm1.SpeedButton18Click(Sender: TObject);

```

```

begin
    if SpeedButton12.Caption=' ' then
        begin
            SpeedButton12.Caption:=SpeedButton18.Caption;
            SpeedButton18.Caption:=' ';
        end
    else
    if SpeedButton15.Caption=' ' then
        begin
            SpeedButton15.Caption:=SpeedButton18.Caption;
            SpeedButton18.Caption:=' ';
        end;
    end;

    procedure TForm1.SpeedButton14Click(Sender: TObject);
    begin
        if SpeedButton13.Caption=' ' then
            begin
                SpeedButton13.Caption:=SpeedButton14.Caption;
                SpeedButton14.Caption:=' ';
            end else
            if SpeedButton10.Caption=' ' then
                begin
                    SpeedButton10.Caption:=SpeedButton14.Caption;
                    SpeedButton14.Caption:=' ';
                end else
                if SpeedButton15.Caption=' ' then
                    begin
                        SpeedButton15.Caption:=SpeedButton14.Caption;
                        SpeedButton14.Caption:=' ';
                    end;
                end;

        procedure TForm1.SpeedButton13Click(Sender: TObject);
        begin
            if SpeedButton9.Caption=' ' then
                begin
                    SpeedButton9.Caption:=SpeedButton13.Caption;
                    SpeedButton13.Caption:=' ';
                end else
                if SpeedButton14.Caption=' ' then
                    begin
                        SpeedButton14.Caption:=SpeedButton13.Caption;
                        SpeedButton13.Caption:=' ';
                    end;
                end;

        procedure TForm1.SpeedButton9Click(Sender: TObject);
        begin
            if SpeedButton5.Caption=' ' then
                begin
                    SpeedButton5.Caption:=SpeedButton9.Caption;
                    SpeedButton9.Caption:=' ';
                end else
                if SpeedButton10.Caption=' ' then
                    begin
                        SpeedButton10.Caption:=SpeedButton9.Caption;
                        SpeedButton9.Caption:=' ';
                    end else
                    if SpeedButton13.Caption=' ' then
                        begin
                            SpeedButton13.Caption:=SpeedButton9.Caption;
                            SpeedButton9.Caption:=' ';
                        end;
                    end;
                end;
            end;
        end;
    end;

```

```

procedure TForm1.SpeedButton10Click(Sender: TObject);
begin
if SpeedButton6.Caption=' ' then
begin
SpeedButton6.Caption:=SpeedButton10.Caption;
SpeedButton10.Caption:=' ';
end else
if SpeedButton9.Caption=' ' then
begin
SpeedButton9.Caption:=SpeedButton10.Caption;
SpeedButton10.Caption:=' ';
end else
if SpeedButton11.Caption=' ' then
begin
SpeedButton11.Caption:=SpeedButton10.Caption;
SpeedButton10.Caption:=' ';
end else
if SpeedButton14.Caption=' ' then
begin
SpeedButton14.Caption:=SpeedButton10.Caption;
SpeedButton10.Caption:=' ';
end;
end;
end;

```

```

procedure TForm1.SpeedButton11Click(Sender: TObject);
begin
if SpeedButton7.Caption=' ' then
begin
SpeedButton7.Caption:=SpeedButton11.Caption;
SpeedButton11.Caption:=' ';
End else
if SpeedButton10.Caption=' ' then
begin
SpeedButton10.Caption:=SpeedButton11.Caption;
SpeedButton11.Caption:=' ';
End else
if SpeedButton12.Caption=' ' then
begin
SpeedButton12.Caption:=SpeedButton11.Caption;
SpeedButton11.Caption:=' ';
End else
if SpeedButton15.Caption=' ' then
begin
SpeedButton15.Caption:=SpeedButton11.Caption;
SpeedButton11.Caption:=' ';
end;
end;
end;

```

```

procedure TForm1.SpeedButton12Click(Sender: TObject);
begin
if SpeedButton8.Caption=' ' then
begin
SpeedButton8.Caption:=SpeedButton12.Caption;
SpeedButton12.Caption:=' ';
End else
if SpeedButton11.Caption=' ' then
begin
SpeedButton11.Caption:=SpeedButton12.Caption;
SpeedButton12.Caption:=' ';
End else
if SpeedButton18.Caption=' ' then
begin
SpeedButton18.Caption:=SpeedButton12.Caption;
SpeedButton12.Caption:=' ';
end;
end;
end;

```

```

procedure TForm1.SpeedButton5Click(Sender: TObject);
begin
if SpeedButton1.Caption=' ' then
begin
SpeedButton1.Caption:=SpeedButton5.Caption;
SpeedButton5.Caption:=' ';
End else
if SpeedButton6.Caption=' ' then
begin
SpeedButton6.Caption:=SpeedButton5.Caption;
SpeedButton5.Caption:=' ';
End else
if SpeedButton9.Caption=' ' then
begin
SpeedButton9.Caption:=SpeedButton5.Caption;
SpeedButton5.Caption:=' ';
end;
end;
end;

```

```

procedure TForm1.SpeedButton6Click(Sender: TObject);
begin
if SpeedButton2.Caption=' ' then
begin
SpeedButton2.Caption:=SpeedButton6.Caption;
SpeedButton6.Caption:=' ';
End else
if SpeedButton5.Caption=' ' then
begin
SpeedButton5.Caption:=SpeedButton6.Caption;
SpeedButton6.Caption:=' ';
End else
if SpeedButton7.Caption=' ' then
begin
SpeedButton7.Caption:=SpeedButton6.Caption;
SpeedButton6.Caption:=' ';
End else
if SpeedButton10.Caption=' ' then
begin
SpeedButton10.Caption:=SpeedButton6.Caption;
SpeedButton6.Caption:=' ';
end;
end;
end;

```

```

procedure TForm1.SpeedButton7Click(Sender: TObject);
begin
if SpeedButton3.Caption=' ' then
begin
SpeedButton3.Caption:=SpeedButton7.Caption;
SpeedButton7.Caption:=' ';
End else
if SpeedButton6.Caption=' ' then
begin
SpeedButton6.Caption:=SpeedButton7.Caption;
SpeedButton7.Caption:=' ';
End else
if SpeedButton8.Caption=' ' then
begin
SpeedButton8.Caption:=SpeedButton7.Caption;
SpeedButton7.Caption:=' ';
End else
if SpeedButton11.Caption=' ' then
begin
SpeedButton11.Caption:=SpeedButton7.Caption;
SpeedButton7.Caption:=' ';
end;
end;
end;

```

```

end;

procedure TForm1.SpeedButton8Click(Sender: TObject);
begin
if SpeedButton4.Caption='' then
begin
SpeedButton4.Caption:=SpeedButton8.Caption;
SpeedButton8.Caption:='';
End else
if SpeedButton7.Caption='' then
begin
SpeedButton7.Caption:=SpeedButton8.Caption;
SpeedButton8.Caption:='';
End else
if SpeedButton12.Caption='' then
begin
SpeedButton12.Caption:=SpeedButton8.Caption;
SpeedButton8.Caption:='';
end;
end;

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
if SpeedButton2.Caption='' then
begin
SpeedButton2.Caption:=SpeedButton1.Caption;
SpeedButton1.Caption:='';
End else
if SpeedButton5.Caption='' then
begin
SpeedButton5.Caption:=SpeedButton1.Caption;
SpeedButton1.Caption:='';
end;
end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
if SpeedButton1.Caption='' then
begin
SpeedButton1.Caption:=SpeedButton2.Caption;
SpeedButton2.Caption:='';
End else
if SpeedButton3.Caption='' then
begin
SpeedButton3.Caption:=SpeedButton2.Caption;
SpeedButton2.Caption:='';
End else
if SpeedButton6.Caption='' then
begin
SpeedButton6.Caption:=SpeedButton2.Caption;
SpeedButton2.Caption:='';
end;

end;

procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
if SpeedButton2.Caption='' then

```

```

begin
SpeedButton2.Caption:=SpeedButton3.Caption;
SpeedButton3.Caption:='';
End else
if SpeedButton4.Caption='' then
begin
SpeedButton4.Caption:=SpeedButton3.Caption;
SpeedButton3.Caption:='';
End else
if SpeedButton7.Caption='' then
begin
SpeedButton7.Caption:=SpeedButton3.Caption;
SpeedButton3.Caption:='';
end;
end;

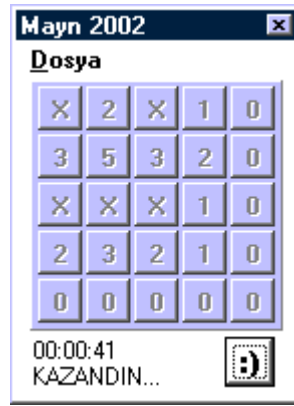
procedure TForm1.SpeedButton4Click(Sender: TObject);
begin
if SpeedButton3.Caption='' then
begin
SpeedButton3.Caption:=SpeedButton4.Caption;
SpeedButton4.Caption:='';
End else
if SpeedButton8.Caption='' then
begin
SpeedButton8.Caption:=SpeedButton4.Caption;
SpeedButton4.Caption:='';
end;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
form1.Caption:='Puzzle - ' + TimeToStr(now-zaman);
if (SpeedButton1.Caption='1')and
(SpeedButton2.Caption='2')
and(SpeedButton3.Caption='3')and
(SpeedButton4.Caption='4')
and(SpeedButton5.Caption='5')and
(SpeedButton6.Caption='6')
and(SpeedButton7.Caption='7')and
(SpeedButton8.Caption='8')
and(SpeedButton9.Caption='9')and
(SpeedButton10.Caption='10')
and(SpeedButton11.Caption='11')and
(SpeedButton12.Caption='12')
and(SpeedButton13.Caption='13')and
(SpeedButton14.Caption='14')
and(SpeedButton15.Caption='15')and
(SpeedButton18.Caption='')
then begin
timer1.Enabled:=false;
form1.Caption:='Puzzle - ' + TimeToStr(now-zaman);
ShowMessage('Finished!..');
end;
end;

end.

```

Sırada Windows ile gelen oyunlardan Mayın Oyunu'na benzer bir oyunumuz var. Tek eksiği boş hücrelerin etrafındaki boş olan diğer hücreleri patlatmamak. Form üzerinde 5x5 yani 25 adet SpeedButton, bir Timer1 ve bir MainMenu1 bulunuyor. SpeedButton'a resim, oyuna da ses efektleri eklenebilir. Süre ölçümü ve buna göre oynayanların rekor sürenin sıralanması da oyunu daha hoş hale getirir. Delphi 5 dili kullanarak yaptığım projeyi CD'de "Ornekler\Delphi\Mayın\mayn.dpr" adı ile bulabilirsiniz. Tüm SpeedButton'ların OnClick olayı SpeedButton1'e yönlendirilmiş durumda. "Form1.Components" komutu için güzel bir uygulama...



Resim 22c – Mayın Oyunumuzun ekran görüntüsü

Kodlar:

```

unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  Menus, ExtCtrls, Buttons, StdCtrls;

type
  TForm1 = class(TForm)
    Panel1: TPanel;
    SpeedButton1: TSpeedButton;
    Timer1: TTimer;
    MainMenu1: TMainMenu;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    SpeedButton5: TSpeedButton;
    SpeedButton6: TSpeedButton;
    SpeedButton7: TSpeedButton;
    SpeedButton8: TSpeedButton;
    SpeedButton9: TSpeedButton;
    SpeedButton10: TSpeedButton;
    SpeedButton11: TSpeedButton;
    SpeedButton12: TSpeedButton;
    SpeedButton13: TSpeedButton;
    SpeedButton14: TSpeedButton;
    SpeedButton15: TSpeedButton;
    SpeedButton16: TSpeedButton;
    SpeedButton17: TSpeedButton;
    SpeedButton18: TSpeedButton;
    SpeedButton19: TSpeedButton;
    SpeedButton20: TSpeedButton;
    SpeedButton21: TSpeedButton;
    SpeedButton22: TSpeedButton;
    SpeedButton23: TSpeedButton;
    SpeedButton24: TSpeedButton;
    SpeedButton25: TSpeedButton;
    Dosya1: TMenuItem;
    Yeni1: TMenuItem;
    Kapat1: TMenuItem;
    Label1: TLabel;
    BitBtn1: TBitBtn;
    procedure Kapat1Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  end;

```

```

    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
  type
    hucre=record
      deger:integer;
      bos:boolean;
    end;

  var
    Form1: TForm1;
    dizi:array[1..25] of hucre;
    sayac,b:byte;
    zaman:TTime;

  implementation

{$R *.DFM}

    procedure tusdoldur;
    var a:integer;
    begin
      repeat
        a:=random(24)+1;
        if dizi[a].bos then
          begin
            dizi[a].bos:=false;
            dizi[a].deger:=a;
            inc(b);
            inc(sayac)
          end
        else
          tusdoldur;
      until (sayac=25) or (b=5);
    end;

    function adrakam(ad:string):integer;
    var gecici:string;i:integer;
    begin
      for i:=length(ad) downto 0 do begin
        if ad[i] in ['0'..'9'] then
          gecici:=ad[i]+gecici;
        end;
      end;
      adrakam:=strtoint(gecici);
    end;

```

```

end;

procedure TForm1.Kapat1Click(Sender: TObject);
begin
close
end;

function kontrol(gecici:string):integer;
var toplam:integer;
begin
    toplam:=0; //komşuların boş olanları kaç adet?
    case strtoint(gecici) of

1:toplam:=ord(dizi[2].deger<>0)+ord(dizi[6].deger<>0)+
    ord(dizi[7].deger<>0);

2:toplam:=ord(dizi[1].deger<>0)+ord(dizi[6].deger<>0)+
    ord(dizi[7].deger<>0)+ord(dizi[8].deger<>0)+
    ord(dizi[3].deger<>0);

3:toplam:=ord(dizi[2].deger<>0)+ord(dizi[4].deger<>0)+
    ord(dizi[7].deger<>0)+ord(dizi[8].deger<>0)+
    ord(dizi[9].deger<>0);

4:toplam:=ord(dizi[3].deger<>0)+ord(dizi[5].deger<>0)+
    ord(dizi[8].deger<>0)+ord(dizi[9].deger<>0)+
    ord(dizi[10].deger<>0);

5:toplam:=ord(dizi[4].deger<>0)+ord(dizi[9].deger<>0)+
    ord(dizi[10].deger<>0);

6:toplam:=ord(dizi[1].deger<>0)+ord(dizi[2].deger<>0)+
    ord(dizi[7].deger<>0)+ord(dizi[11].deger<>0)+
    ord(dizi[12].deger<>0);

7:toplam:=ord(dizi[1].deger<>0)+ord(dizi[2].deger<>0)+
    ord(dizi[3].deger<>0)+ord(dizi[6].deger<>0)+
    ord(dizi[11].deger<>0)+ord(dizi[12].deger<>0)+
    ord(dizi[13].deger<>0)+
    ord(dizi[8].deger<>0);

8:toplam:=ord(dizi[2].deger<>0)+ord(dizi[3].deger<>0)+
    ord(dizi[4].deger<>0)+ord(dizi[7].deger<>0)+
    ord(dizi[9].deger<>0)+ord(dizi[12].deger<>0)+
    ord(dizi[13].deger<>0)+
    ord(dizi[14].deger<>0);

9:toplam:=ord(dizi[3].deger<>0)+ord(dizi[4].deger<>0)+
    ord(dizi[5].deger<>0)+ord(dizi[8].deger<>0)+
    ord(dizi[10].deger<>0)+ord(dizi[15].deger<>0)+
    ord(dizi[13].deger<>0)+
    ord(dizi[14].deger<>0);

10:toplam:=ord(dizi[4].deger<>0)+ord(dizi[5].deger<>0)+
    ord(dizi[9].deger<>0)+ord(dizi[14].deger<>0)+
    ord(dizi[15].deger<>0);

11:toplam:=ord(dizi[6].deger<>0)+ord(dizi[7].deger<>0)+
    ord(dizi[12].deger<>0)+ord(dizi[16].deger<>0)+
    ord(dizi[17].deger<>0);

12:toplam:=ord(dizi[6].deger<>0)+ord(dizi[7].deger<>0)+
    ord(dizi[8].deger<>0)+ord(dizi[11].deger<>0)+
    ord(dizi[13].deger<>0)+ord(dizi[16].deger<>0)+
    ord(dizi[17].deger<>0)+
    ord(dizi[18].deger<>0);

```

```

13:toplam:=ord(dizi[7].deger<>0)+ord(dizi[8].deger<>0)+
    ord(dizi[9].deger<>0)+ord(dizi[12].deger<>0)+
    ord(dizi[14].deger<>0)+ord(dizi[17].deger<>0)+
    ord(dizi[18].deger<>0)+
    ord(dizi[19].deger<>0);

14:toplam:=ord(dizi[8].deger<>0)+ord(dizi[9].deger<>0)+
    ord(dizi[10].deger<>0)+ord(dizi[13].deger<>0)+
    ord(dizi[15].deger<>0)+ord(dizi[18].deger<>0)+
    ord(dizi[19].deger<>0)+
    ord(dizi[20].deger<>0);

15:toplam:=ord(dizi[9].deger<>0)+ord(dizi[10].deger<>0)+
    ord(dizi[14].deger<>0)+ord(dizi[19].deger<>0)+
    ord(dizi[20].deger<>0);

16:toplam:=ord(dizi[11].deger<>0)+ord(dizi[12].deger<>0)+
    ord(dizi[17].deger<>0)+ord(dizi[21].deger<>0)+
    ord(dizi[22].deger<>0);

17:toplam:=ord(dizi[11].deger<>0)+ord(dizi[12].deger<>0)+
    ord(dizi[13].deger<>0)+ord(dizi[16].deger<>0)+
    ord(dizi[18].deger<>0)+ord(dizi[21].deger<>0)+
    ord(dizi[22].deger<>0)+
    ord(dizi[23].deger<>0);

18:toplam:=ord(dizi[12].deger<>0)+ord(dizi[13].deger<>0)+
    ord(dizi[14].deger<>0)+ord(dizi[17].deger<>0)+
    ord(dizi[19].deger<>0)+ord(dizi[22].deger<>0)+
    ord(dizi[23].deger<>0)+
    ord(dizi[24].deger<>0);

19:toplam:=ord(dizi[13].deger<>0)+ord(dizi[14].deger<>0)+
    ord(dizi[15].deger<>0)+ord(dizi[18].deger<>0)+
    ord(dizi[20].deger<>0)+ord(dizi[23].deger<>0)+
    ord(dizi[24].deger<>0)+
    ord(dizi[25].deger<>0);

20:toplam:=ord(dizi[14].deger<>0)+ord(dizi[15].deger<>0)+
    ord(dizi[19].deger<>0)+ord(dizi[24].deger<>0)+
    ord(dizi[25].deger<>0);

21:toplam:=ord(dizi[16].deger<>0)+ord(dizi[17].deger<>0)+
    ord(dizi[22].deger<>0);

22:toplam:=ord(dizi[16].deger<>0)+ord(dizi[17].deger<>0)+
    ord(dizi[18].deger<>0)+ord(dizi[21].deger<>0)+
    ord(dizi[23].deger<>0);

23:toplam:=ord(dizi[17].deger<>0)+ord(dizi[18].deger<>0)+
    ord(dizi[19].deger<>0)+ord(dizi[22].deger<>0)+
    ord(dizi[24].deger<>0);

24:toplam:=ord(dizi[18].deger<>0)+ord(dizi[19].deger<>0)+
    ord(dizi[20].deger<>0)+ord(dizi[23].deger<>0)+
    ord(dizi[25].deger<>0);

25:toplam:=ord(dizi[19].deger<>0)+ord(dizi[20].deger<>0)+
    ord(dizi[24].deger<>0);

    end;
    kontrol:=toplam;
end;

procedure TForm1.SpeedButton1Click(Sender: TObject);
var ad,gecici:string;i,sil:integer;

```

```

begin
if timer1.Enabled=false then begin
  timer1.Enabled:=true;
  zaman:=now();
end;
ad:=(sender as Tspeedbutton).name;
//basılan tuşun numarasını al...
for i:=length(ad) downto 0 do begin
  if ad[i] in ['0'..'9'] then
    gecici:=ad[i]+gecici;
end;

  if dizi[strtoint(gecici)].deger=0 then
//mayına mı bastın?
  begin
    if kontrol(gecici)=0 then //patlat...
      begin
        //under construction !?!
      end;
    (sender as
TSpeedButton).caption:=inttostr(kontrol(gecici));
    MessageBeep(1111);
    sil:=0;
    for i:=0 to ComponentCount-1 do
//kaç tane boş kaldı?
      if components[i] is TSpeedButton then
        if (components[i] as TSpeedButton).caption="" then
inc(sil);
        if sil=5 then begin
          label1.caption:=label1.caption+#13+'KAZANDIN...';
          timer1.Enabled:=false;
          for i:=0 to ComponentCount-1 do //iş bitti
            begin
              if components[i] is TSpeedButton then begin
                (components[i] as TSpeedButton).enabled:=false;
                sil:=adrakam( (components[i] as
TSpeedButton).name);
                if dizi[sil].bos=false then (components[i] as
TSpeedButton).caption:='X';
              end;
            end;
          end;
        else
          begin

```

```

(sender as TSpeedButton).caption:='X';
MessageBeep(11);
for i:=0 to ComponentCount-1 do
  begin
    if components[i] is TSpeedButton then begin
      (components[i] as TSpeedButton).enabled:=false;
      sil:=adrakam( (components[i] as
TSpeedButton).name);
      if dizi[sil].bos=false then (components[i] as
TSpeedButton).caption:='X';
    end;
  end;
  timer1.Enabled:=false;
  label1.caption:='Kaybettin...';
end;

end;

procedure TForm1.FormCreate(Sender: TObject);
var i:integer;
begin
  randomize;
  b:=0;
  sayac:=0;
  label1.Caption:='';
  for i:=1 to 25 do begin
    dizi[i].bos:=true;
    dizi[i].deger:=0;
  end;
  tUSDoldur;
  for i:=0 to ComponentCount-1 do
    begin
      if components[i] is TSpeedButton then begin
        (components[i] as TSpeedButton).caption:='';
        (components[i] as TSpeedButton).enabled:=true;
      end;
    end;
  end;

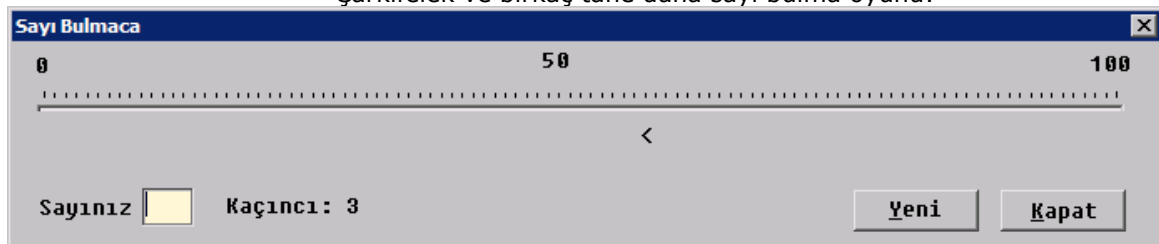
  procedure TForm1.Timer1Timer(Sender: TObject);
  begin
    label1.caption:=timetostr(now()-zaman);
  end;

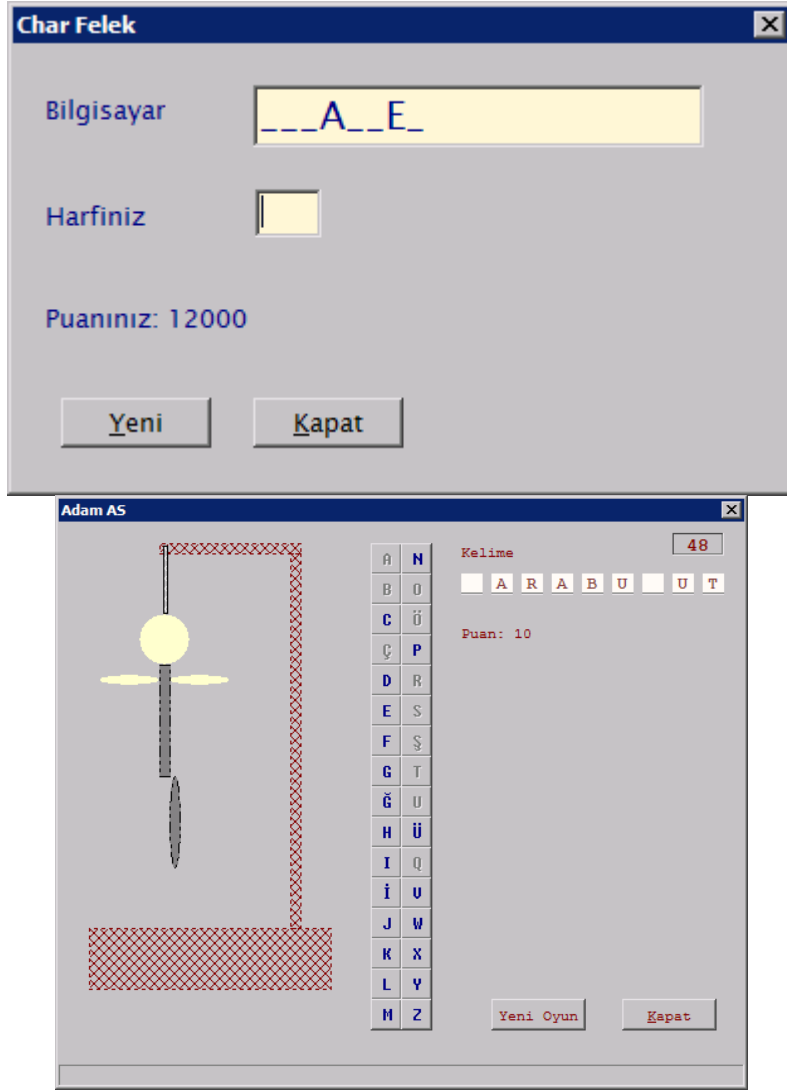
end.

```

Dikkat ederseniz 3x3, 4x4 ve 5x5 hücreler halinde yapabileceğimiz oyunlardan son oyundaki hücre sayısının 5x5 ile en fazla olmasına rağmen kod satır sayısı en az. Öğrenmeye çalıştığımız dile ne kadar hakim olabilirsek o kadar güzel programlar yazabiliriz. "Diger\Extras\codebankv2.zip" ile kurabileceğiniz Delphi Türk Kod Bankası programındaki sayıyı yazıya çevirme algoritmalarına bakarsanız, birden fazla çeşit olduklarını görebilirsiniz. Hem kısa hem de güvenilir kod üretmemiz gereklidir. Bu arada hız için güvenlikten ödün vermek birçok böceğe sebep olabilir. Hala bizim göremediğimiz böcekler, program içinde kullanıcılar için ortaya çıkabilir. Tek çare bol bol tasarımı denemektir.

Çarkifelek ve birkaç tane daha sayı bulma oyunu:



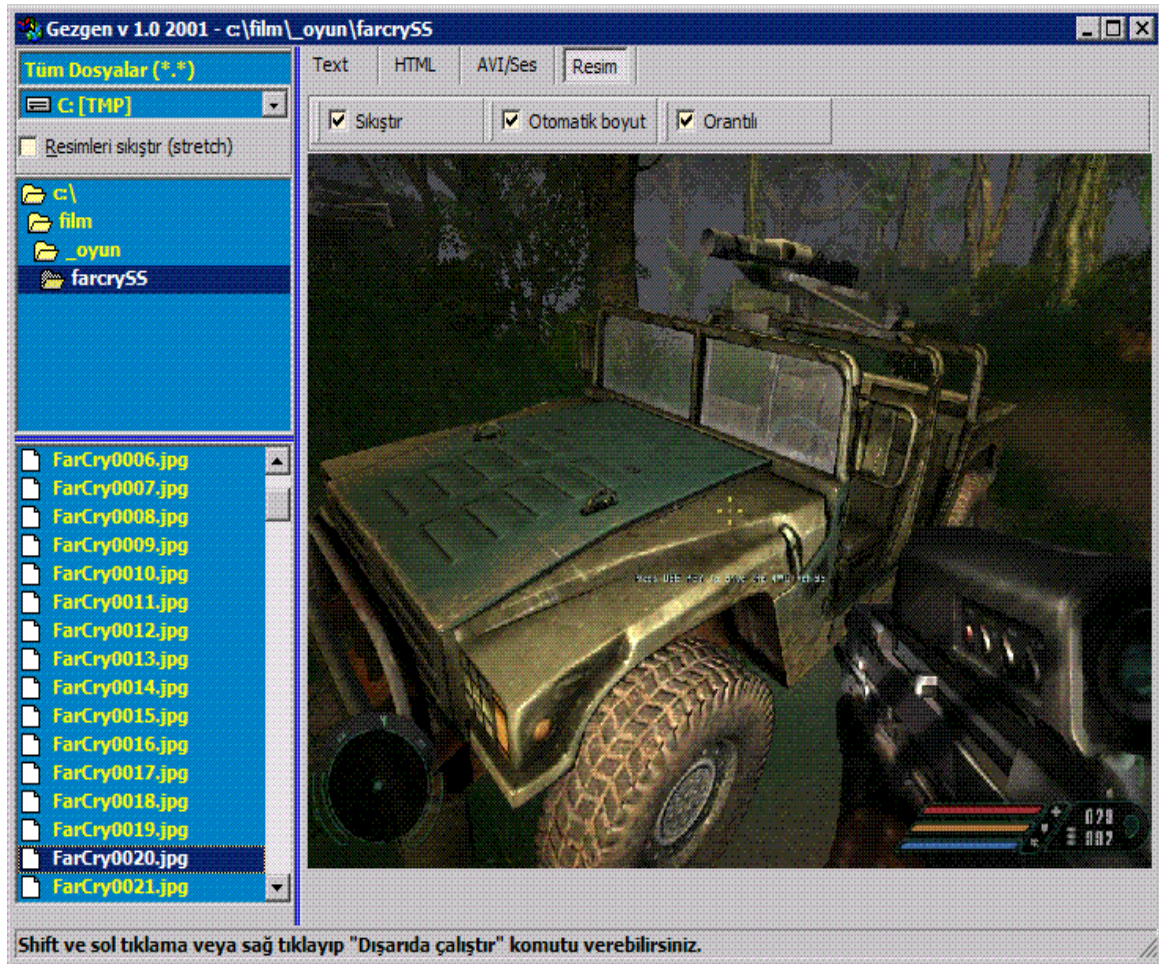


Resim 22d – Diğer oyunlar

15. Gezgen (Windows Gezgini yapımı)

(Ornekler\Delphi\gezgen\Project2.dpr)

Metin, HTML, *.avi ve resim dosyalarını gösteren, Windows Gezgini'nin basit hali. *.tga, *.gif gibi dosyaları Image nesnesi desteklemediği halde isterseniz dosyaya sağ tıklayıp "HTML" komutu verebilir ve izleyebilirsiniz. Fakat Internetten bulabileceğiniz Delphi nesneleri ile diğer dosya uzantılarına da (*.mp3, *.mov, *.mpg...) destek sağlayabilirsiniz. Bir eksik de bul menüsünün olmaması. Belli kriterde arama yaparak o dosyaları sağ tarafta kullanıcıya sunsak iyi olurdu, Delphi örneklerinden Dosyalistesi adlı program da kullanılabilir. "Ornekler\Delphi\Gezgin2\Project1.dpr" ile her iki programımızın birleşmiş halini yaptık, incelemenizi tavsiye ederim.



Resim 23 – Basit bir gezgin çalışması işte...

Nesne ve kodlarımız:

object Form1: TForm1

Left = 279

Top = 140

Width = 579

Height = 481

Caption = 'Gezgen v 1.0 2001'

Color = clBtnFace

Font.Charset = DEFAULT_CHARSET

Font.Color = clWindowText

Font.Height = -11

Font.Name = 'MS Sans Serif'

Font.Style = []

OldCreateOrder = False

OnCreate = FormCreate

PixelsPerInch = 96

TextHeight = 13

object Splitter1: TSplitter

Left = 177

Top = 0

Width = 5

Height = 433

Cursor = crHSplit

Beveled = True

Color = clScrollBar

ParentColor = False

end

object PageControl1: TPageControl

Left = 182

```

Top = 0
Width = 387
Height = 433
ActivePage = TabSheet1
Align = alClient
Style = tsFlatButtons
TabOrder = 0
object TabSheet1: TTabSheet
  Caption = 'Text '
  object RichEdit1: TRichEdit
    Left = 0
    Top = 0
    Width = 379
    Height = 402
    Align = alClient
    BorderStyle = bsNone
    ScrollBars = ssBoth
    TabOrder = 0
  end
end
object TabSheet2: TTabSheet
  Caption = 'HTML'
  ImageIndex = 1
  object WebBrowser1: TWebBrowser
    Left = 0
    Top = 0
    Width = 379
    Height = 421
    Align = alClient
    TabOrder = 0
    ControlData = {
      ...
    }
  end
end
object TabSheet3: TTabSheet
  Caption = 'AVI/Ses'
  ImageIndex = 2
  object Panel2: TPanel
    Left = 0
    Top = 41
    Width = 379
    Height = 380
    Align = alClient
    BevelOuter = bvNone
    Color = clWindow
    TabOrder = 0
  end
  object ToolBar1: TToolBar
    Left = 0
    Top = 0
    Width = 379
    Height = 41
    ButtonHeight = 30
    Caption = 'ToolBar1'
    TabOrder = 1
    object MediaPlayer1: TMediaPlayer
      Left = 0
      Top = 2
      Width = 253
      Height = 30
      Display = Panel2
      TabOrder = 0
    end
  end
end
end

```

```

object TabSheet5: TTabSheet
  Caption = 'Resim'
  ImageIndex = 4
  object Image1: TImage
    Left = 0
    Top = 0
    Width = 379
    Height = 421
    Align = alClient
  end
end
object Panel1: TPanel
  Left = 0
  Top = 0
  Width = 177
  Height = 433
  Align = alLeft
  BevelOuter = bvNone
  TabOrder = 1
  object Splitter2: TSplitter
    Left = 0
    Top = 218
    Width = 177
    Height = 5
    Cursor = crVSplit
    Align = alTop
    Beveled = True
    Color = clScrollBar
    ParentColor = False
  end
  object FileListBox1: TFileListBox
    Left = 0
    Top = 223
    Width = 177
    Height = 196
    Hint =
      'Shift ve sol tıklayıp dışarıya dosyayı açabilirsiniz...|Shift
      ve' +
      ' sol tıklama veya sağ tıklayıp "Dışarıda çalıştır" komutu
      verebi' +
      'lirsiniz.'
    Align = alClient
    Color = 12615680
    FileType = [ftReadOnly, ftHidden, ftSystem, ftVolumeID,
      ftArchive, ftNormal]
    Font.Charset = TURKISH_CHARSET
    Font.Color = clYellow
    Font.Height = -11
    Font.Name = 'Tahoma'
    Font.Style = [fsBold]
    IntegralHeight = True
    ItemHeight = 16
    ParentFont = False
    ParentShowHint = False
    PopupMenu = PopupMenu1
    ShowGlyphs = True
    ShowHint = True
    TabOrder = 0
    OnClick = FileListBox1Click
    OnMouseDown = FileListBox1MouseDown
  end
  object DirectoryListBox1: TDirectoryListBox
    Left = 0
    Top = 73
    Width = 177

```

```

Height = 145
Align = alTop
Color = 12615680
FileList = FileListBox1
Font.Charset = TURKISH_CHARSET
Font.Color = clYellow
Font.Height = -11
Font.Name = 'Tahoma'
Font.Style = [fsBold]
ItemHeight = 16
ParentFont = False
TabOrder = 1
OnChange = DirectoryListBox1Change
end
object CoolBar1: TCoolBar
Left = 0
Top = 0
Width = 177
Height = 73
BandBorderStyle = bsNone
Bands = <
item
Break = False
Control = FilterComboBox1
FixedSize = True
ImageIndex = -1
MinHeight = 21
Width = 327
end
item
Break = False
Control = DriveComboBox1
FixedSize = True
ImageIndex = -1
MinHeight = 19
Width = 173
end
item
Break = False
Control = CheckBox1
FixedSize = True
ImageIndex = -1
MinHeight = 29
Width = 173
end>
object FilterComboBox1: TFilterComboBox
Left = 0
Top = 0
Width = 327
Height = 21
Hint = 'Dosya listesini sınırlandırma'
Color = 12615680
FileList = FileListBox1
Filter =
'Tüm Dosyalar (*.*)|*.txt|Text (*.txt;*.rtf;*.ini;*.bat)|*.txt;*.rt
+
'f;*.ini;*.bat|HTML (*.htm;*.html)|*.htm;*.html|AVI
(*.avi)|*.avi' +
'Ses (*.wav;*.mid)|*.wav;*.mid|Resim
(*.bmp;*.jpg;*.jpeg;*.emf;*.wmf;*.dib;*.ico)|*.bmp;*.jpg;*.jpeg;*.emf;*.wmf;*.dib;*.ico'
Font.Charset = TURKISH_CHARSET
Font.Color = clYellow
Font.Height = -11
Font.Name = 'Tahoma'

```

```

Font.Style = [fsBold]
ParentFont = False
TabOrder = 0
end
object DriveComboBox1: TDriveComboBox
Left = 0
Top = 21
Width = 173
Height = 19
Hint = 'Sürücü seçimi'
Color = 12615680
DirList = DirectoryListBox1
Font.Charset = TURKISH_CHARSET
Font.Color = clYellow
Font.Height = -11
Font.Name = 'Tahoma'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 1
TextCase = tcUpperCase
end
object CheckBox1: TCheckBox
Left = 0
Top = 40
Width = 173
Height = 29
Hint =
'Resim dosyalarının tam ekran yapma veya orjinal
boyutunda bırakm' +
'a komutu'
Caption = '&Resimleri sıkıştır (stretch)'
ParentShowHint = False
ShowHint = True
TabOrder = 2
OnClick = CheckBox1Click
end
end
object StatusBar1: TStatusBar
Left = 0
Top = 433
Width = 569
Height = 19
Panels = <
item
Text = 'Selam...'
Width = 50
end>
SimplePanel = False
end
object PopupMenu1: TPopupMenu
Left = 232
Top = 147
object DosyayAadakiTiplereZorla1: TMenuItem
Caption = 'Dosyayı Aşağıdaki Tiplere Zorla'
Enabled = False
end
object Text1: TMenuItem
Caption = '&Text'
Hint = 'Seçili dosya metin olarak açılmaya zorlanır'
OnClick = Text1Click
end
object HTML1: TMenuItem
Caption = '&HTML'
Hint = 'Seçili dosya web sayfası olarak açılmaya zorlanır'
OnClick = HTML1Click

```

```

end
object AVISes1: TMenuItem
  Caption = '&AVI/Ses'
  Hint = 'Seçili dosya ses veya film olarak açılmaya zorlanır'
 OnClick = AVISes1Click
end
object Resim1: TMenuItem
  Caption = '&Resim'
  Hint = 'Seçili dosya resim olarak açılmaya zorlanır'
  OnClick = Resim1Click
end

```

```

object N1: TMenuItem
  Caption = '-'
end
object Daryaalt1: TMenuItem
  Caption = '&Dışarıda çalıştır'
  Hint = 'Windows içinde kayıtlı dosya türü olarak açma
komutu'
  OnClick = Daryaalt1Click
end
end
end

```

```
unit Unit1; //-----kodlar, program buradan başlıyor-----
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
Tabs, ComCtrls, ExtCtrls, StdCtrls, FileCtrl, MPlayer, ToolWin, OleCtrls,
SHDocVw, Menus, jpeg;
```

```
type
```

```

TForm1 = class(TForm)
  AVISes1: TMenuItem;
  CheckBox1: TCheckBox;
  CoolBar1: TCoolBar;
  Daryaalt1: TMenuItem;
  DirectoryListBox1: TDirectoryListBox;
  DosyayAadakiTiplereZorla1: TMenuItem;
  DriveComboBox1: TDriveComboBox;
  FileListBox1: TFileListBox;
  FilterComboBox1: TFilterComboBox;
  HTML1: TMenuItem;
  Image1: TImage;
  MediaPlayer1: TMediaPlayer;
  N1: TMenuItem;
  PageControl1: TPageControl;
  Panel1: TPanel;
  Panel2: TPanel;
  PopupMenu1: TPopupMenu;
  Resim1: TMenuItem;
  RichEdit1: TRichEdit;
  Splitter1: TSplitter;
  Splitter2: TSplitter;
  StatusBar1: TStatusBar;
  TabSheet1: TTabSheet;
  TabSheet2: TTabSheet;
  TabSheet3: TTabSheet;
  TabSheet5: TTabSheet;
  Text1: TMenuItem;
  ToolBar1: TToolBar;
  WebBrowser1: TWebBrowser; //Delphi 3'de HTML nesnesi bulunuyor...
  procedure DirectoryListBox1Change(Sender: TObject);
  procedure FileListBox1Click(Sender: TObject);
  procedure FileListBox1MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
  procedure Text1Click(Sender: TObject);
  procedure HTML1Click(Sender: TObject);
  procedure AVISes1Click(Sender: TObject);
  procedure Resim1Click(Sender: TObject);
  procedure CheckBox1Click(Sender: TObject);
  procedure Daryaalt1Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure ShowHint(Sender: TObject);
  function uzantibul:string;
private

```

```

    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;

implementation
{$R *.DFM}

function TForm1.uzantibul:string;
var uzanti:string;i:byte;dosyaadi:string;
begin
i:=strlen(pchar(FileListBox1.FileName));
dosyaadi:=lowercase(FileListBox1.FileName);
uzanti:="";
    while (dosyaadi[i]<>'.' ) do
        begin
            uzanti:=dosyaadi[i]+uzanti;
            dec(i);
        end;
uzantibul:=uzanti;
end;

procedure TForm1.FileListBox1Click(Sender: TObject);
var
    Flags: OLEVariant;
begin
caption:= 'Gezgen v 1.0 2001 - '+FileListBox1.Directory;
if (uzantibul='txt') or (uzantibul='rtf') or (uzantibul='bat') or (uzantibul='ini') then
begin
    PageControl1.ActivePageIndex:=0;
    RichEdit1.Lines.LoadFromFile(FileListBox1.FileName);
end else
if (uzantibul='htm') or (uzantibul='html') then
begin
    PageControl1.ActivePageIndex:=1;
    WebBrowser1.Navigate(WideString(FileListBox1.FileName), Flags, Flags, Flags, Flags);
    //HTML1.RequestDoc('file:///'+ FileListBox1.FileName); Delphi 3 için gerekli kod böyle olabilir... 3 adet / kullanımına dikkat ediniz.
end else
if (uzantibul='avi') or (uzantibul='wav') or (uzantibul='mid') then
begin
    PageControl1.ActivePageIndex:=2;
    panel2.Repaint;
    MediaPlayer1.AutoOpen:=false;
    MediaPlayer1.Close;
    MediaPlayer1.FileName:=FileListBox1.FileName;
    MediaPlayer1.AutoOpen:=true;
    MediaPlayer1.Open;
    MediaPlayer1.Play;
end else
if (uzantibul='dib') or (uzantibul='jpg') or (uzantibul='jpeg') or (uzantibul='bmp') or (uzantibul='ico') or (uzantibul='wmf') or
(uzantibul='emf') then
begin
    PageControl1.ActivePageIndex:=3;
    Image1.Picture.LoadFromFile(FileListBox1.FileName);
end else
begin
caption:= 'Gezgen v 1.0 2001 - '+FileListBox1.Directory + ' - Pardon, Dosya izlenemiyor!';
end;
end;

procedure TForm1.DirectoryListBox1Change(Sender: TObject);
begin caption:= 'Gezgen v 1.0 2001 - '+FileListBox1.Directory; end;

```

```

procedure TForm1.FileListBox1MouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var dizin1:string;dizin2:pchar;
begin
if ssShift in Shift then
begin
  dizin1:='start '"+FileListBox1.FileName+'"; //kırmızı olanlar çift tırnaktır, XP için start yerine explorer yazınız.
  dizin2:=pchar(dizin1);
  winexec(dizin2,0);
end;
end;

procedure TForm1.Text1Click(Sender: TObject);
begin
if FileListBox1.FileName <> " then begin
  PageControl1.ActivePageIndex:=0;
  RichEdit1.Lines.LoadFromFile(FileListBox1.FileName);
end;
end;

procedure TForm1.HTML1Click(Sender: TObject);
var
  Flags: OLEVariant;
begin
if FileListBox1.FileName <> " then begin
  PageControl1.ActivePageIndex:=1;
  WebBrowser1.Navigate(WideString(FileListBox1.FileName), Flags, Flags, Flags, Flags);
  //HTML1.RequestDoc('file:/// + FileListBox1.FileName); //Delphi 3 için gerekli kod böyle olabilir...
end;
end;

procedure TForm1.AVISes1Click(Sender: TObject);
begin
if FileListBox1.FileName <> " then begin
  PageControl1.ActivePageIndex:=2;
  panel2.Repaint;
  MediaPlayer1.AutoOpen:=false;
  MediaPlayer1.Close;
  MediaPlayer1.FileName:=FileListBox1.FileName;
  MediaPlayer1.AutoOpen:=true;
  MediaPlayer1.Open; MediaPlayer1.Play;
end;end;

procedure TForm1.Resim1Click(Sender: TObject);
begin
if FileListBox1.FileName <> " then begin //iki bitişik tek tırnak
  PageControl1.ActivePageIndex:=3; Image1.Picture.LoadFromFile(FileListBox1.FileName);
end; end;

procedure TForm1.CheckBox1Click(Sender: TObject);
begin image1.Stretch:=CheckBox1.Checked; end;

procedure TForm1.Daryaalt1Click(Sender: TObject);
var dizin1:string;dizin2:pchar;
begin
  dizin1:='start '"+FileListBox1.FileName+'"; //buradaki tek tırnakların sayıları sıra ile 1, 3 ve 4 adet olacak
  dizin2:=pchar(dizin1); winexec(dizin2,0);
end;

procedure TForm1.FormCreate(Sender: TObject);
begin Application.OnHint:=ShowHint; Application.HintColor:=claquea; end;

procedure TForm1.ShowHint(Sender: TObject);
begin statusbar1.simpleText:=Application.Hint; end;
end.

```

16. Dosya işlemi (Write ve Read komutları ile dosyalama)

(Ornekler\Delphi\dosya\Project1.dpr)

Pascal dilindeki gibi Writeln ve Readln komutları ile metin dosya oluşturuyoruz. Memo nesnesine LoadFromFile ve SaveToFile komutlarını kullanmadan da satır ekleme ve kaydetme yapabiliriz. DOS programcılığında bu yöntem sık kullanılıyordu, Delphi'de ise çok kolay. CD örneğinde anlattığım gibi *.ini, metin tabanlı dosya, veritabanı ve registry defterine kayıt imkanlarımız vardır. Başka bir bilgisayarda çalıştırmak için BDE'nin de taşınmasına gerek kalmıyor.

Bu örnekte sadece bir boyut var, o da satır. Tablolamanın en basit hali. Veri arama, değiştirme, sıralama, filtreleme gibi üst seviye veritabanı olaylarını eklemek oldukça uzun kod yazmamıza bağlı ve bu kodlar da hataya çok açık. Yani güvencelerle çalışmak en iyisi olabilir:

try

```
dosya:=OpenDialog1.FileName;
caption:= 'Dosya İşlemleri - '+dosya;
memo1.Clear;
AssignFile(f, dosya);
Reset(F);
while not Eof(f) do
begin Readln(f, satir); memo1.Lines.Add(satir); end;
```

finally

```
CloseFile(f);
```

end;

VB'deki listbox örneği gibi biz burada da Memo yerine Listbox kullanılabiliriz. Memo1 nesnesinin bir avantajı LoadFromFile ve SaveToFile gibi komutlarıdır, fakat burada sadece Write ve Read komutlarını kullanmak zorundayız.



Resim 24 – Basit dosya işlemleri

```
object Form1: TForm1
  Left = 271
  Top = 104
  BorderStyle = bsDialog
  Caption = 'Dosya İşlemleri - varsayılan.txt'
  ClientHeight = 320
  ClientWidth = 364
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
```

```
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  OldCreateOrder = False
  PixelsPerInch = 96
  TextHeight = 13
  object Button1: TButton
    Left = 24
```



```

Top = 16
Width = 75
Height = 25
Caption = '&Aç'
TabOrder = 0
OnClick = Button1Click
end
object Memo1: TMemo
Left = 8
Top = 80
Width = 345
Height = 233
Lines.Strings = (
'Memo1 - salt okunur...')
ReadOnly = True
TabOrder = 1
end
object Button3: TButton
Left = 256
Top = 16
Width = 75
Height = 25
Caption = 'K&apat'
TabOrder = 2
OnClick = Button3Click
end
object Edit1: TEdit
Left = 16
Top = 50
Width = 121
Height = 21
TabOrder = 3
end

```

```

object Button4: TButton
Left = 152
Top = 48
Width = 75
Height = 25
Caption = '&Satır Ekle'
TabOrder = 4
OnClick = Button4Click
end
object Button2: TButton
Left = 101
Top = 16
Width = 75
Height = 25
Caption = '&Farklı Kaydet'
TabOrder = 5
OnClick = Button2Click
end
object OpenFileDialog1: TOpenDialog
Filter = 'Tüm Dosyalar|*.*|Text (*.txt)|*.txt'
Title = 'Diş - Aç'
Left = 88
Top = 112
end
object SaveDialog1: TSaveDialog
DefaultExt = 'txt'
Filter = 'Tüm Dosyalar|*.*|Text (*.txt)|*.txt'
Title = 'Diş - Kaydet'
Left = 168
Top = 112
end
end

```

Kodlarımız:

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;
type
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Edit1: TEdit;
    Memo1: TMemo;
    OpenFileDialog1: TOpenDialog;
    SaveDialog1: TSaveDialog;
    procedure Button1Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
f: TextFile;
satir:string;

```



```

dosya:string= 'varsayilan.txt';

implementation
{$R *.DFM}
procedure TForm1.Button1Click(Sender: TObject);
begin
  if OpenFileDialog1.Execute then
  begin
    dosya:=OpenDialog1.FileName;
    caption:= 'Dosya İşlemleri - '+dosya;
    memo1.Clear;
    AssignFile(f, dosya);
    Reset(F);                      //okumak için açıyoruz
    while not Eof(f) do
    begin
      ReadLn(f, satir);
      memo1.Lines.Add(satir);
    end;
    CloseFile(f);
  end;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin close; end;

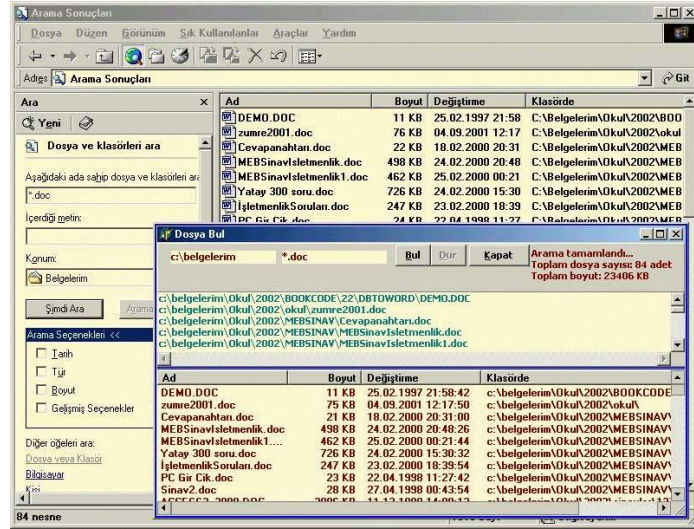
procedure TForm1.Button4Click(Sender: TObject);
begin
  if dosya <> 'varsayilan.txt' then begin
    AssignFile(f, dosya);
    Append(f);                     //kayıt eklemek için açıyoruz
    WriteLn(f, edit1.text);
    Flush(f);
    CloseFile(f);
    memo1.Clear;
    AssignFile(f, dosya);
    Reset(F);
    while not Eof(f) do
    begin
      ReadLn(f, satir);
      memo1.Lines.Add(satir);
    end;
    CloseFile(f);
  end;
end;

procedure TForm1.Button2Click(Sender: TObject);
var i:integer;
begin
  if SaveDialog1.Execute then
  begin
    dosya:= SaveDialog1.FileName;
    caption:= 'Dosya İşlemleri - '+dosya;
    AssignFile(f, dosya);
    Rewrite(f);                   //ilk kez dosya oluşturuyoruz
    for i:=0 to memo1.Lines.Count-1 do
    begin
      WriteLn(f, memo1.lines[i]);
      Flush(f);
    end;
    CloseFile(f);
  end;
end;
end.

```

17. Dosya Listele (dosya bulma işlemleri)

(Ornekler\Delphi\dosyalistele\Project1.dpr)



Resim 25 – Windows ME arama ekranı ve bizim arama programımız

DelphiTurk'den indirdiğim kodlar ile Delphi 5 dilinde hazırladığım bu örneği, Windows'un "Dosya Bul" programı ile karşılaştırdırca biraz daha hızlı olduğunu görebiliriz. En azından çok uzun aramalarda kullanıcı Dur tuşu ile aramayı kesebiliyoruz. Bunun için ProcessMessages komutu ile form canlı tutuluyor. Arama bitince de kaç dosya bulunduğunu ve toplam kaç kilo byte olduğu Label1'e aktarılıyor. *.* filtresi ile bir klasörün kaç KiloByte olduğunu da bulabilirsiniz. Memo1 nesnesi yerine ListBox veya ListView gibi nesneler konulup, daha hoş listeler elde edilebilir. İsterseniz elinizdeki CD içeriklerini, Memo1 nesnesine sağ tıklayıp Kaydet komutu sayesinde ayrı bir dosyaya kaydedip, bir dosyayı bulmak için o CD'yi daha sonra takıp çıkarmaktan kurtulabilirsiniz. Windows komutlarını kullanarak da istediğiniz klasörü kopyalama, silme, taşıma, karşılaştırma gibi özellikler de ekleyebilirsiniz. Yedekleme yapılan klasördeki dosyaları güncelleme amaçlı da olabilir.

object Form1: TForm1

Left = 250

Top = 132

Width = 633

Height = 539

Caption = 'Dosya Bul'

Color = clBtnFace

KeyPreview = True

OldCreateOrder = False

Position = poScreenCenter

OnCreate = FormCreate

OnKeyDown = FormKeyDown

PixelsPerInch = 96

TextHeight = 13

object Splitter1: TSplitter

Left = 0

Top = 217

Width = 623

Height = 5

Cursor = crVSplit

Align = alTop

Beveled = True

Color = clBlue

ParentColor = False

end

object Memo1: TMemo

Left = 0

Top = 57

Width = 623

Height = 160

Align = alTop

BorderStyle = bsNone

ParentFont = False

PopupMenu = PopupMenu2

ScrollBars = ssBoth

TabOrder = 0

WordWrap = False

end

object ListView1: TListView

Left = 0

Top = 222

Width = 623

Height = 288

Align = alClient

BorderStyle = bsNone

Color = 13695231

Columns = <

item

Caption = 'Ad'

Width = 150

end

item

Alignment = taRightJustify

Caption = 'Boyut'

Width = 75

end

```

item
  Caption = 'Değiştirme'
  Width = 120
end
item
  Caption = 'Klasörde'
  Width = 200
end>
FlatScrollBars = True
FullDrag = True
HotTrack = True
HotTrackStyles = [htHandPoint]
HoverTime = 10
IconOptions.AutoArrange = True
Items.Data =
{1D0000000100000000000000FFFFFFFFFFFFFFFF00000000
00000000}
  ParentFont = False
  ParentShowHint = False
  PopupMenu = PopupMenu1
  ShowHint = True
  SortType = stData
  TabOrder = 1
  ViewStyle = vsReport
  OnColumnClick = ListView1ColumnClick
  OnCompare = ListView1Compare
end
object Panel1: TPanel
  Left = 0
  Top = 0
  Width = 623
  Height = 57
  Align = alTop
  TabOrder = 2
  object Label1: TLabel
    Left = 432
    Top = 5
    Width = 85
    Height = 13
    Caption = 'Arama durumu'
  end
  object Button2: TButton
    Left = 360
    Top = 2
    Width = 70
    Height = 25
    Caption = '&Kapat'
    TabOrder = 0
   OnClick = Button2Click
  end
  object Button3: TButton
    Left = 316
    Top = 2
    Width = 41
    Height = 25
    Caption = '&Dur'
    Enabled = False
    TabOrder = 1
   OnClick = Button3Click
  end
  object Button1: TButton
    Left = 275
    Top = 2
    Width = 41
    Height = 25
    Caption = '&Bul'

```

```

  TabOrder = 2
  OnClick = Button1Click
end
object Edit2: TEdit
  Left = 144
  Top = 9
  Width = 121
  Height = 15
  BorderStyle = bsNone
  TabOrder = 3
  Text = '*.doc'
  OnKeyDown = Edit2KeyDown
end
object Edit1: TEdit
  Left = 16
  Top = 9
  Width = 121
  Height = 15
  Hint =
  'Çift tıklayıp başka bir yer seçebilirsiniz, sürücü seçmek
  için i' +
  'se metin kutusuna d: gibi yazıp çift tıklayınız...'
  BorderStyle = bsNone
  ParentShowHint = False
  ShowHint = True
  TabOrder = 4
  Text = 'c:\belgelerim'
  OnDbClick = Edit1DbClick
end
end
object PopupMenu1: TPopupMenu
  Left = 240
  Top = 104
  object BykSimge1: TMenuItem
    Caption = '&Büyük Simge '
    GroupIndex = 1
    RadioItem = True
    OnClick = BykSimge1Click
  end
  object KkSimge1: TMenuItem
    Caption = '&Küçük Simge'
    GroupIndex = 1
    RadioItem = True
    OnClick = KkSimge1Click
  end
  object Liste1: TMenuItem
    Caption = '&Liste'
    GroupIndex = 1
    RadioItem = True
    OnClick = Liste1Click
  end
  object Ayrntlar1: TMenuItem
    Caption = '&Ayrıntılar'
    Checked = True
    Default = True
    GroupIndex = 1
    RadioItem = True
    OnClick = Ayrntlar1Click
  end
  object Grid1: TMenuItem
    Break = mbBarBreak
    Caption = '&Grid'
    GroupIndex = 1
    OnClick = Grid1Click
  end
end
end

```

```

object PopupMenu2: TPopupMenu
  Left = 328
  Top = 104
  object Kaydet1: TMenuItem
    Caption = 'Kaydet'
    OnClick = Kaydet1Click
  end
end
end

```

Kod kısmı:

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, TLHelp32, ComCtrls, ExtCtrls,
  Buttons, ShlObj, ShellAPI, CommCtrl, Clipbrd, Menus,
  ToolWin, ImgList, filectrl;
//SelectDirectory diyalogu için filectrl gerekli

```

```

type
  TForm1 = class(TForm)
    Ayrntlar1: TMenuItem;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    BykSimge1: TMenuItem;
    Edit1: TEdit;
    Edit2: TEdit;
    Grid1: TMenuItem;
    Kaydet1: TMenuItem;
    KkSimge1: TMenuItem;
    Label1: TLabel;
    Liste1: TMenuItem;
    ListView1: TListView;
    Memo1: TMemo;
    Panel1: TPanel;
    PopupMenu1: TPopupMenu;
    PopupMenu2: TPopupMenu;
    SaveDialog1: TSaveDialog;
    Splitter1: TSplitter;
    procedure Button1Click(Sender: TObject);
    procedure FindFile(StartDir, FileMask: string);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure BykSimge1Click(Sender: TObject);
    procedure KkSimge1Click(Sender: TObject);
    procedure Liste1Click(Sender: TObject);
    procedure Ayrntlar1Click(Sender: TObject);
    procedure Grid1Click(Sender: TObject);
    procedure Kaydet1Click(Sender: TObject);
    procedure ListView1ColumnClick(Sender: TObject;
      Column: TListColumn);
    procedure ListView1Compare(Sender: TObject; Item1,
      Item2: TListItem;
        Data: Integer; var Compare: Integer);
    procedure FormCreate(Sender: TObject);
    procedure Edit1DbClick(Sender: TObject);
    procedure Edit2KeyDown(Sender: TObject; var Key:
      Word;
        Shift: TShiftState);
    procedure FormKeyDown(Sender: TObject; var Key:
      Word;
        Shift: TShiftState);
    procedure InitLVImageList(ListView: TListView);
  private

```

```

object SaveDialog1: TSaveDialog
  DefaultExt = 'txt'
  Filter = 'Metin Dosyası|*.txt'
  Title = 'Liste Kaydet'
  Left = 152
  Top = 104
end
end
end

```

```

  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  dosyasayisi: integer;
  boyut: longword;
  yeter: boolean;
  ColumnToSort: Integer; yon: byte;

```

implementation

```

{$R *.dfm}
function GetShellImage(szFileName: string): Integer;
var
  FileInfo: TShellFileInfo;
begin
  SHGetFileInfo(PChar(szFileName), 0, FileInfo,
    SizeOf(FileInfo), SHGFI_SYSICONINDEX or
    SHGFI_USEFILEATTRIBUTES);
  Result := FileInfo.ilcon;
end;

```

```

procedure TForm1.InitLVImageList(ListView: TListView);
var
  FileInfo: TShellFileInfo;
  ImageListHandle: THandle;
begin
  ImageListHandle := SHGetFileInfo('C:\', 0, FileInfo,
    SizeOf(FileInfo), SHGFI_SYSICONINDEX or
    SHGFI_SMALLICON);
  SendMessage(ListView.Handle, LVM_SETIMAGELIST,
    LVSIL_SMALL, ImageListHandle);

  ImageListHandle := SHGetFileInfo('C:\', 0, FileInfo,
    SizeOf(FileInfo), SHGFI_SYSICONINDEX or
    SHGFI_LARGEICON);

  SendMessage(ListView.Handle, LVM_SETIMAGELIST,
    LVSIL_NORMAL, ImageListHandle);
end;

```

```

procedure TForm1.FindFile(StartDir, FileMask: string);
var
  SR: TSearchRec;
  DirList: TStringList;
  IsFound: Boolean;
  i: integer; ListItem: TListItem; tarih: variant;
begin
  if StartDir[length(StartDir)] <> '\' then
    StartDir := StartDir + '\';

```

```

IsFound := FindFirst(StartDir+FileMask, faAnyFile-
faDirectory, SR) = 0;
while IsFound and not(yeter)do begin
  application.processMessages;
  Memo1.Lines.Add(StartDir + SR.Name);
  with ListView1 do begin
    ListItem:= items.Add;
    ListItem.Caption:=sr.Name;
    ListItem.ImageIndex := GetShellImage(sr.Name);
    ListItem.SubItems.Add(format('%10d KB',[(sr.size) div
1024]));
    tarih:=FileDateToDateTime(sr.Time);
    ListItem.SubItems.Add(DateTimeToStr(tarih));
    ListItem.SubItems.Add(StartDir);
  end;
  Inc(dosyasayisi);
  boyut:=boyut+sr.Size;
  IsFound := FindNext(SR) = 0;
end;
FindClose(SR);

DirList := TStringList.Create;
IsFound := FindFirst(StartDir+'*.*', faAnyFile, SR) = 0;
while IsFound and not(yeter)do begin
  if ((SR.Attr and faDirectory) <> 0) and
    (SR.Name[1] <> '.') then
    DirList.Add(StartDir + SR.Name);
  IsFound := FindNext(SR) = 0;
end;
FindClose(SR);

for i := 0 to DirList.Count-1 do
  FindFile(DirList[i], FileMask);
DirList.Free;
end;

procedure TForm1.Button1Click(Sender: TObject);
var kb:real;
begin
memo1.Clear;
ListView1.Items.Clear;
  yeter:=false;
button1.Enabled:=false;
button3.Enabled:=true;
dosyasayisi:=0;
boyut:=0;
FindFile(edit1.Text,edit2.Text);
kb:=boyut/1024;
label1.Caption:='Arama tamamlandı...'+#13+'Toplam dosya
sayısı: '+IntToStr(dosyasayisi)+' adet'
      +#13+'Toplam boyut: '+IntToStr(round(kb))+ ' KB';
button1.Enabled:=true;
button3.Enabled:=false;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
close
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
yeter:=true;
button1.Enabled:=true;
button3.Enabled:=false;
end;

```

```

procedure TForm1.BykSimge1Click(Sender: TObject);
begin
BykSimge1.Checked:=true;
ListView1.ViewStyle:=vsIcon;
end;

```

```

procedure TForm1.KkSimge1Click(Sender: TObject);
begin
kkSimge1.Checked:=true;
ListView1.ViewStyle:=vsSmallIcon;
end;

```

```

procedure TForm1.Liste1Click(Sender: TObject);
begin
Liste1.Checked:=true;
ListView1.ViewStyle:=vsList;
end;

```

```

procedure TForm1.Ayrntlar1Click(Sender: TObject);
begin
Ayrntlar1.Checked:=true;
ListView1.ViewStyle:=vsReport;
end;

```

```

procedure TForm1.Grid1Click(Sender: TObject);
begin
Grid1.Checked:=not(Grid1.Checked);
ListView1.GridLines:=Grid1.Checked;
end;

```

```

procedure TForm1.Kaydet1Click(Sender: TObject);
begin
if SaveDialog1.Execute then
Memo1.Lines.SaveToFile(SaveDialog1.FileName);
end;

```

```

procedure TForm1.ListView1ColumnClick(Sender:
TObject;
  Column: TListColumn);
begin
ColumnToSort := Column.Index;
(Sender as TCustomListView).AlphaSort;
yon:=-1*yon;
end;

```

```

procedure TForm1.ListView1Compare(Sender: TObject;
Item1, Item2: TListItem;
  Data: Integer; var Compare: Integer);
var
  ix: Integer;
begin
if ColumnToSort = 0 then
  Compare :=
yon+CompareText(Item1.Caption,Item2.Caption)
else begin
  ix := ColumnToSort - 1;
  Compare :=
yon+CompareText(Item1.SubItems[ix],Item2.SubItems[ix]);
end;
end;

```

```

procedure TForm1.FormCreate(Sender: TObject);
begin
yon:=1;
InitLVImageList(ListView1);

```

```
end;

procedure TForm1.Edit1DbClick(Sender: TObject);
var result:string;
begin
if SelectDirectory('Klasör Seç', Edit1.Text, Result) then
Edit1.Text := Result;
end;

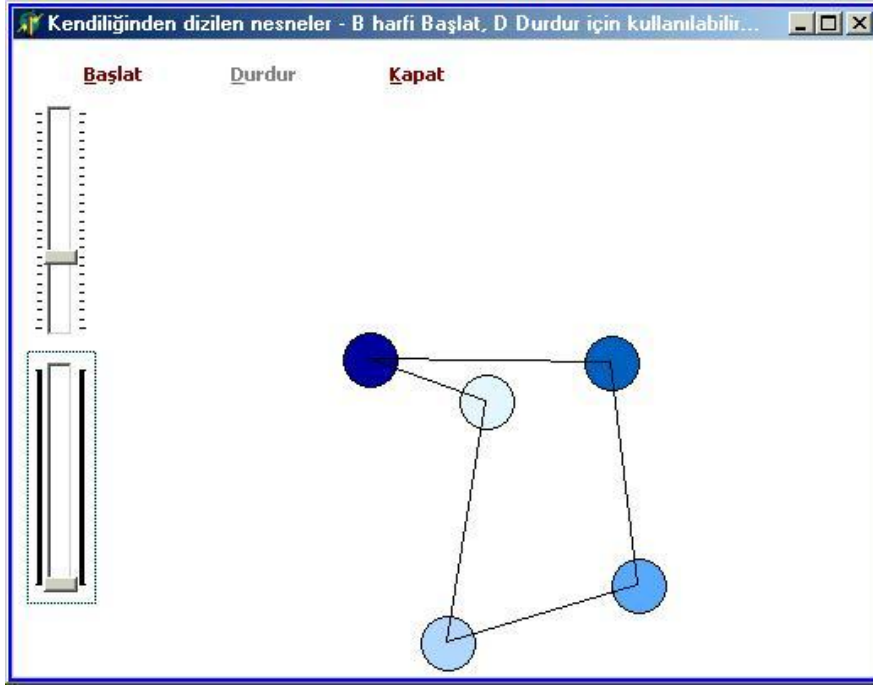
procedure TForm1.Edit2KeyDown(Sender: TObject; var Key:
Word;
  Shift: TShiftState);
begin
  if key=VK_RETURN then Button1Click(self); //bul
end;

procedure TForm1.FormKeyDown(Sender: TObject; var
Key: Word;
  Shift: TShiftState);
begin
if key=VK_ESCAPE then Button3Click(self); //dur
end;

end.
```

18. think2 (kendiliğinden dizilen nesneler)

(Ornekler\Delphi\think2\Project2.dpr)



Resim 26 – Otomatik olarak dizilen nesneler

Bu örnekteki amaç birbirlerine olan uzaklığı ölçerek, en yakın topa yaklaşmaya çalışan nesnelerdir. Bu denemeler sonucunda dairemsi olarak sıralanmaları gerekiyordu. Fakat her zaman renk dizilimleri koyudan açığa doğru bize göre pek olmuyor. Çünkü aralarındaki takipçi çizgilere bakarsanız bazen birbiri üzerilerine geçip çakışıyorlar. Tam olarak hedefim aralıkları eşit oluncaya dek birbirlerine doğru ilerlemeleri ve beş topun daire gibi olmaları idi. Java Development Kit 1.2.2'nin demo dizininde GraphLayout adlı bir örnekte bizim örneğin tam çalışmasını görebilirsiniz.

En kısa yol problemi ile ilk kez VB dilinde askerdayken uğraştığım WinLines, adlı oyunumda karşılaştım. Basit 9x9 boyutlarındaki grid - ızgara dizilerinde bile bir topun hedefe en kısa hangi yoldan gidebileceği beni bayağı zorlamıştı. Aslında hala bu problemi tam olarak bilemiyorum. Bilgisayar oyunlarının çoğunda bu soruna benzer, cisimlerin belli bir yere en insansı, akıllı gibi karar vererek gitmeleri bulunur. Artificial Intelligence - yapay zeka en güçlü makineleri bile zorlayan büyük programcılık gücü gerektiren bir konudur. Yine de bu konuda biraz kafa yormak ve sonuçta bir şeyler elde etmek çok hoşuma gidiyor. Hatıralarsanız TicTocToe örneğinde de bilgisayara sıra gelince hamlesini düşünmeyi öğretmek gerekmişti.

Nesnelerimiz:

```
object Form1: TForm1
  Left = 271
  Top = 104
  Width = 528
  Height = 396
  Caption =
    'Kendiliğinden dizilen nesneler - B harfi Başlat, D
    Durdur için k' +
    'ullanılabilir...'
  Color = clWhite
  Font.Charset = TURKISH_CHARSET
  Font.Color = clMaroon
  Font.Height = -11
  Font.Name = 'Tahoma'
```

```
Font.Style = [fsBold]
OldCreateOrder = False
Position = poScreenCenter
OnCreate = FormCreate
PixelsPerInch = 96
TextHeight = 13
object Shape1: TShape
  Left = 96
  Top = 168
  Width = 33
  Height = 33
  Brush.Color = 10485760
  Shape = stCircle
end
```

```

object Shape2: TShape
  Left = 160
  Top = 168
  Width = 33
  Height = 33
  Brush.Color = 13001472
  Shape = stCircle
end
object Shape4: TShape
  Left = 288
  Top = 168
  Width = 33
  Height = 33
  Brush.Color = 16767411
  Shape = stCircle
end
object Shape5: TShape
  Left = 352
  Top = 168
  Width = 33
  Height = 33
  Brush.Color = 16775658
  Shape = stCircle
end
object SpeedButton1: TSpeedButton
  Left = 15
  Top = 8
  Width = 89
  Height = 22
  Hint = 'Hız'
  Caption = '&Başlat'
  Flat = True
  ParentShowHint = False
  ShowHint = True
  OnClick = SpeedButton1Click
end
object SpeedButton2: TSpeedButton
  Left = 104
  Top = 8
  Width = 89
  Height = 22
  Caption = '&Durdur'
  Flat = True
  OnClick = SpeedButton2Click
end
object Shape3: TShape
  Left = 224
  Top = 168
  Width = 33
  Height = 33
  Brush.Color = 16756059
  Shape = stCircle
end
object SpeedButton3: TSpeedButton
  Left = 195
  Top = 8
  Width = 89

```

```

  Height = 22
  Caption = '&Kapat'
  Flat = True
  OnClick = SpeedButton3Click
end
object TrackBar1: TTrackBar
  Left = 8
  Top = 32
  Width = 41
  Height = 150
  Hint = 'Hız (1-25 birim), aşağısı daha büyük rakamdır...'
  Max = 25
  Min = 1
  Orientation = trVertical
  ParentShowHint = False
  Frequency = 1
  Position = 15
  SelEnd = 0
  SelStart = 0
  ShowHint = True
  TabOrder = 0
  TickMarks = tmBoth
  TickStyle = tsAuto
  OnChange = TrackBar1Change
end
object TrackBar2: TTrackBar
  Left = 8
  Top = 184
  Width = 41
  Height = 150
  Hint =
    'Topların aralıkları (1-150 piksel), aşağısı daha büyük
    rakamdır.' +
    '..'
  Max = 150
  Min = 1
  Orientation = trVertical
  ParentShowHint = False
  Frequency = 1
  Position = 100
  SelEnd = 0
  SelStart = 0
  ShowHint = True
  TabOrder = 1
  TickMarks = tmBoth
  TickStyle = tsAuto
  OnChange = TrackBar2Change
end
object Timer1: TTimer
  Enabled = False
  Interval = 250
  OnTimer = Timer1Timer
  Left = 72
  Top = 72
end
end

```

Kodlarımız:

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  Buttons, ExtCtrls, ComCtrls;

```

```

type
  TForm1 = class(TForm)
    Shape1: TShape;
    Shape2: TShape;
    Shape3: TShape;
    Shape4: TShape;

```



```

Shape5: TShape;
SpeedButton1: TSpeedButton;
SpeedButton2: TSpeedButton;
SpeedButton3: TSpeedButton;
Timer1: TTimer;
TrackBar1: TTrackBar;
TrackBar2: TTrackBar;
procedure SpeedButton1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure TrackBar1Change(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure TrackBar2Change(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
devam: boolean;
hiz, aralik : integer;
baslangic, fark: TDateTime;
implementation
{$R *.DFM}
procedure TForm1.SpeedButton1Click(Sender: TObject);
var mesafe: integer;
begin
  mesafe:=66;
  shape1.Top:=random(height-mesafe);
  shape2.Top:=random(height-mesafe);
  shape3.Top:=random(height-mesafe);
  shape4.Top:=random(height-mesafe);
  shape5.Top:=random(height-mesafe);
  shape1.left:=random(width-mesafe);
  shape2.left:=random(width-mesafe);
  shape3.left:=random(width-mesafe);
  shape4.left:=random(width-mesafe);
  shape5.left:=random(width-mesafe);
  devam:=true;
  SpeedButton2.Enabled:=true;
  SpeedButton1.Enabled:=false;
  baslangic:=now();
  timer1.Enabled:=true;
  //kendiliklerinden dizilme komutu gibi
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  randomize;
  timer1.Enabled:=false;
  hiz:=TrackBar1.Position; aralik:=TrackBar2.Position;
  SpeedButton2.Enabled:=false;
end;

procedure TForm1.SpeedButton3Click(Sender: TObject);
begin close; end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
  timer1.Enabled:=false;
  SpeedButton1.Enabled:=true;
  SpeedButton2.Enabled:=false;
end;

```

```

procedure TForm1.TrackBar1Change(Sender: TObject);
begin hiz:=TrackBar1.Position; end;

procedure TForm1.TrackBar2Change(Sender: TObject);
begin aralik:=TrackBar2.Position; end;

procedure TForm1.Timer1Timer(Sender: TObject);
var  a1,b1,a2,b2,c1,c2:integer;
    deger11,deger12,deger21,deger22:integer;
    hareket_bitti:integer;
begin
  form1.refresh;
  shape5.Top:=round (height/2);
  //son top uzay referansidir
  shape5.left:=round (width/2);
  //:=width DIV 2; de kullanilabilir
  hareket_bitti:=0;
  with form1.Canvas do begin
    a1:=shape2.left-shape1.left; //1 2
    b1:=shape2.top-shape1.top;
    c1:=round (sqrt(a1*a1+b1*b1));
    //dik üçgen formülü, c1 hipotenüs
    a2:=shape5.left-shape1.left; //1 5
    b2:=shape5.top-shape1.top;
    c2:=round (sqrt(a2*a2+b2*b2));
    deger11:= round(a1/hiz);
    deger12:= round(b1/hiz);
    deger21:= round(a2/hiz);
    deger22:= round(b2/hiz);

    if c1>=c2 then begin
      if c1>=aralik then begin
        shape1.left:=shape1.left+deger11;
        shape1.top:=shape1.top+deger12;
      end;
    end else begin
      if c2>=aralik then begin
        shape1.left:=shape1.left+deger21;
        shape1.top:=shape1.top+deger22;
      end;
    end;

    if (c1<aralik) and (c2<aralik) then
      inc(hareket_bitti);

    a1:=shape3.left-shape2.left; //2 3
    b1:=shape3.top-shape2.top;
    c1:=round (sqrt(a1*a1+b1*b1));
    //dik üçgen formülü, c1 hipotenüs anlamındadır
    a2:=shape2.left-shape1.left; //1 2
    b2:=shape2.top-shape1.top;
    c2:=round (sqrt(a2*a2+b2*b2));
    deger11:= round(a1/hiz);
    deger12:= round(b1/hiz);
    deger21:= round(a2/hiz);
    deger22:= round(b2/hiz);

    if c1>=c2 then begin
      if c1>=aralik then begin
        shape2.left:=shape2.left+deger11;
        shape2.top:=shape2.top+deger12;
      end;
    end else begin
      if c2>=aralik then begin
        shape2.left:=shape2.left+deger21;
        shape2.top:=shape2.top+deger22;
      end;
    end;
  end;
end;

```

```

end;
end;

if (c1<aralik) and (c2<aralik) then
inc(hareket_bitti);

a1:=shape4.left-shape3.left; //3 4
b1:=shape4.top-shape3.top;
c1:=round (sqrt(a1*a1+b1*b1));
//dik üçgen formülü, c1 hipotenüs
a2:=shape3.left-shape2.left; //2 3
b2:=shape3.top-shape2.top;
c2:=round (sqrt(a2*a2+b2*b2));
deger11:= round(a1/hiz);
deger12:= round(b1/hiz);
deger21:= round(a2/hiz);
deger22:= round(b2/hiz);

if c1>=c2 then begin
if c1>=aralik then begin
shape3.left:=shape3.left+deger11;
shape3.top:=shape3.top+deger12;
end;
end else begin
if c2>=aralik then begin
shape3.left:=shape3.left+deger21;
shape3.top:=shape3.top+deger22;
end;
end;

if (c1<aralik) and (c2<aralik) then
inc(hareket_bitti);

a1:=shape5.left-shape4.left; //4 5
b1:=shape5.top-shape4.top;
c1:=round (sqrt(a1*a1+b1*b1));
//dik üçgen formülü, c1 hipotenüs
a2:=shape4.left-shape3.left; //3 4
b2:=shape4.top-shape3.top;

```

```

c2:=round (sqrt(a2*a2+b2*b2));
deger11:= round(a1/hiz);
deger12:= round(b1/hiz);
deger21:= round(a2/hiz);
deger22:= round(b2/hiz);

if c1>=c2 then begin
if c1>=aralik then begin
shape4.left:=shape4.left+deger11;
shape4.top:=shape4.top+deger12;
end;
end else begin
if c2>=aralik then begin
shape4.left:=shape4.left+deger21;
shape4.top:=shape4.top+deger22;
end;
end;
if (c1<aralik) and (c2<aralik) then
inc(hareket_bitti);
if hareket_bitti=4 then
begin
SpeedButton2.Click; fark:=now()-baslangic;
Application.MessageBox(pchar('Siralama bitti:
'+timetostr(fark)), 'İşlem tamam', 0);
end;
MoveTo(shape1.left+15,shape1.top+15);
LineTo(shape2.left+15,shape2.top+15);
MoveTo(shape2.left+15,shape2.top+15);
LineTo(shape3.left+15,shape3.top+15);
MoveTo(shape3.left+15,shape3.top+15);
LineTo(shape4.left+15,shape4.top+15);
MoveTo(shape4.left+15,shape4.top+15);
LineTo(shape5.left+15,shape5.top+15);
MoveTo(shape5.left+15,shape5.top+15);
LineTo(shape1.left+15,shape1.top+15);
end;
end;
end.

```

19. DirectX ve OpenGL programcılığı (grafik programcılığı)

(Ornekler\Delphi\3Dfx dizini)

Kitabın konuları içinde en ağır olanı bu kısmı. Delphi dili Quake veya Half – Life oyunları benzeri, endüstriyel robot tasarımı veya tıp alanında görsel tasarım oluşturmak için ideal... DelphiX ve GLScene nesnelerini tanıtarak basit iki örnek yapacağız. Karşılaştığım sorunları görerek siz daha iyilerini tabii ki yapabilirsiniz. 3Dfx ile 1998 yılında Voodoo Banshee ekran kartım varken uğraşmışım, ama artık ATI ve GeForce ekran kartları çok ucuz ve piyasaya hakim. Daha yeni ekran kartınız olabilir, muhakkak DirectX ve OpenGL destekli olmalı. Microsoft DirectX kodları ile daha hızlı donanıma (ses kartı, modem, ekran kartı, joystick...) erişen programlar yazılabilirken, OpenGL'in görüntü kalitesi ve yetenekleri daha fazla. Bu arada oynadığınız oyunların kaliteleri sadece ekran kartının gelişmişliğine değil işlemci, RAM, sabit disk, anakarta da bağlıdır.

En azından elimizde Delphi3 olmalıdır. Delphi 3, 4 ve 5'i destekleyen bu nesneleri "Diger\Component" dizininde bulabilirsiniz. Delphi 6 ve üstü sürümler için ise internetten araştırarak elde edilebilir. *.dpk uzantılı dosyalarına çift tıklayarak kurabiliriz. Örneğin Delphi 5 için DelphiX 2000 şöyle kurulabilir: *.zip dosyamızı bir klasöre açarız, "DelphiX\Source" dizinindeki DelphiX_for5.dpk dosyasını çift tıklar veya Delphi 5'den "File*Open.." ile de bu dosyayı açabiliriz. Karşıımıza çıkan Package - DelphiX_for5.dpk penceresindeki araç çubuğundan Compile komutunu veririz. Ardından Install tıklandığında karşıımıza hangi nesnelerin eklendiğini gösteren uyarı gelir. Component Palette'in en sonunda DelphiX'i görebilirsiniz. Bir arkadaşımız bu nesnelerden sadece ses için gerekeni programlarında müzik çalmak için kullanıyordu. Demek ki tüm nesneleri kullanmak zorunda değiliz. Son olarak da "File*Close All" dediğimizde karşıımıza çıkan soruda, DelphiX nesnelerini Kaydet dememiz gerekiyor.

glscene.org sitesinden indirdiğiniz GLScene de bu şekilde kurulabilir. İsterseniz başka bir yol olan şu şekilde de bu nesne paketini kuralım: Delphi'yi açıp, "File*Open...*GLScene5.dpk" komutu verelim. Compile ve Install deyince artık nesne listemizde GLScene de var olacaktır. Eğer eskiden bazı nesne kurma denemeleriniz olup artikları kalmışsa hata mesajları ile karşılaşabilirsiniz. Tek yapmanız gereken onları nesne listesinden sağ tıklayıp silmeniz.

Her iki pakette de bol örnek var. Uçak ve araba simülasyonları güzel. Yine de sitelerine giderek yenilikleri ve gelişmiş örnekleri bilgisayarınıza indirebilirsiniz. Tutorial – öğretici ve Sample – örnekler bir şeyler yapabilmemiz için genellikle yeterli. Size "Ornekler\ Delphi\ 3Dfx\ DX1\project1.dpr" ve "Ornekler\ Delphi \3Dfx \ Move \ActorTwoCam.dpr" örneklerinden bahsedeceğim. "Ornekler \Delphi \3Dfx \tut2.dpr" sadece 3Dfx kartı desteklediği için hata mesajı verebilir. Ama eğer kodlarını incellerseniz, istenen boyutta (orantılanarak) çizgilerle dijital rakam yazma, kayan yıldızlar, dönen bir altıgen, çözünürlük değiştirme, frekans değiştirme gibi hoş kodlar bulabilirsiniz.

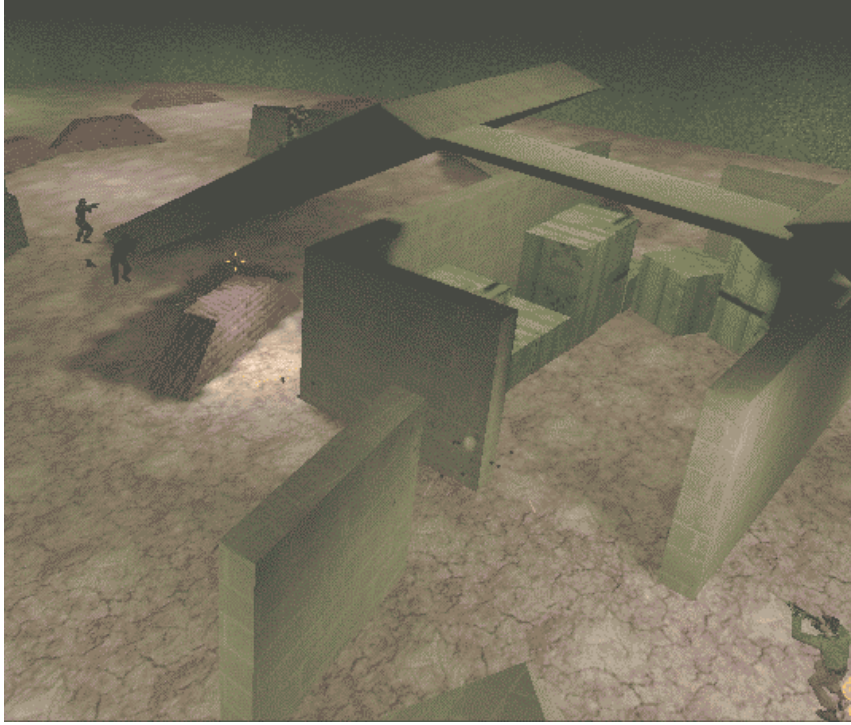
Bu arada yine de biraz 3DSMax karıştırmış olmak gerekiyor. Çoğu yerde *.3ds ve *.x uzantılı dosyalarla karşılaşırız. *.md2 ise Quake2 aktör dosyasıdır. Sadece aktörler değil onların silahları da *.md2 formatındadır.

Elimizde basit de olsa bir materyal editörü ve harita editörü gerekli. Sadece Delphi'de maalesef aradığımızı bulamıyoruz. Terragen, Photoshop gibi programlarla yeryüzü şekilleri oluşturmak, Lightwave veya Studio Max ile aktör ve malzeme hazırlamak üst seviyeye çıkmasak da karşılaştığımız adımlar. Bir sokak veya ev tasarımı, müzik, efektler, yangın, duman, sis, hareketli – animasyonlu nesneler (kırırdayan ağaçlar, şelale, ışık parlamaları, parçalanabilen nesneler), su, ışık, ayna, bulutların hareket ettiği gökyüzü, etkileşimli nesneler (düğme, kol, mayın, tuzaklar, asansör, kapı, pencere, cam kırılması, delinebilen duvarlar, kırılabilen tahta, sandık), kamera açıları, zoom – büyütme / küçültme, kar, yağmur, çamur, yer çekimi, uzay gibi bir çok aşılması gereken problem var. Bunları aslında eski DOS oyunlarında bile görüyoruz. Öyleyse artık dünyada bazı standartlar oluşmuş olmalıdır. Tekrar Amerika'yı keşfetmek yerine internetten aradığınız çeşitte ve yöntemde bilgiye ulaşabilirsiniz.

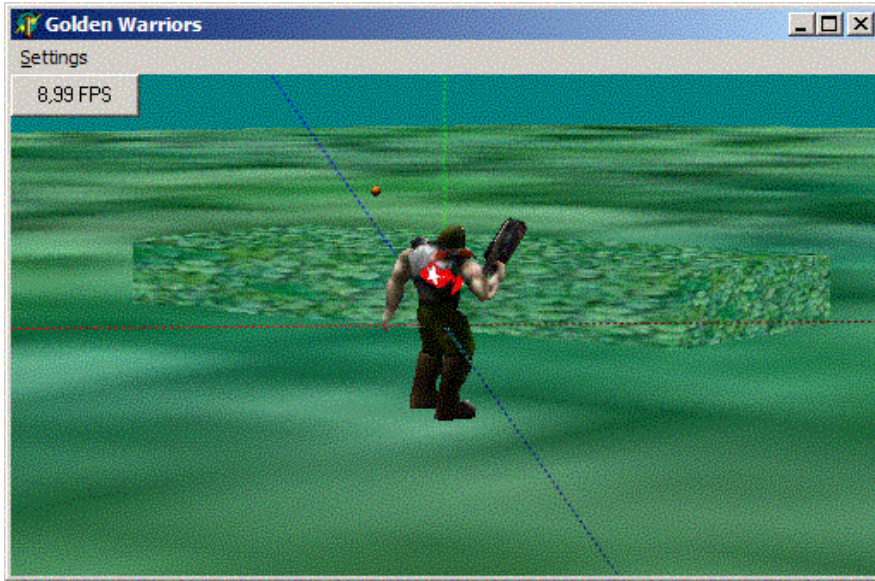
En azından *.wrl yani VRML – Virtual Reality Markup Language (sanal gerçeklik dili) ile uğraştığınızda basit collision – çarpma, yer çekimi, cam, açılabilir kapılar, düğmeler, madde tasarımı, aktör oluşturmak, hikaye, zaman çizgisi, sahne, kare (Macromedia Flash'da da artık iki ve üç boyutlu tasarımlar yapabiliriz) gibi deyimler daha tanıdık hale gelebilir. "CD2\3d\VRML\ISA10.ZIP, ISB30.ZIP, ICA3.ZIP" dosyalarını kurarak Internet Explorer'da üç boyutlu dünyaya kavuşabilirsiniz. CORTVRML.EXE ile VRML plug-in – eklentisini elde ediyoruz. Daha profesyonel editörler de arayabilirsiniz. spazz_v24_full.exe programını da tavsiye ederim. Studio Max'de de Export menüsünden yaptığınız çalışmalarını *.wrl olarak kaydedebilirsiniz. AutoCAD'de tasarlanmış nesneleri de *.wrl biçimine dönüştürebilirsiniz.

Bir önerim de, 3DSMax gibi üst seviye bir programa geçmeden önce basit editörleri kullanarak üçüncü boyuta ısınmaktır. Örneğin Quake, Half – Life Counter Strike için modifikasyonlar yani haritalar yapabiliriz. Bazı oyun CD'lerinde kendi editörleri ile karşılaşabilirsiniz. Bulamazsanız internette muhakkak düşük boyutlu editör programları bulabilirsiniz. "CD2\3d\Diger\worldcraft33.zip" ile basit CS haritaları üretebilirsiniz. Program tasarım dosyanızı

3DSMax benzeri pencerelerde hazırlamanızı ve *.bsp, *.map uzantılı dosyalara dönüştürme işini yapıyor. "Ornekler\Diger\cs_maps.zip" ile yaptığım bazı haritaları görebilirsiniz.



Resim 26 – HL-CS için de_tersane haritam (kasım 2001)



Resim 27 – GLScene örneği

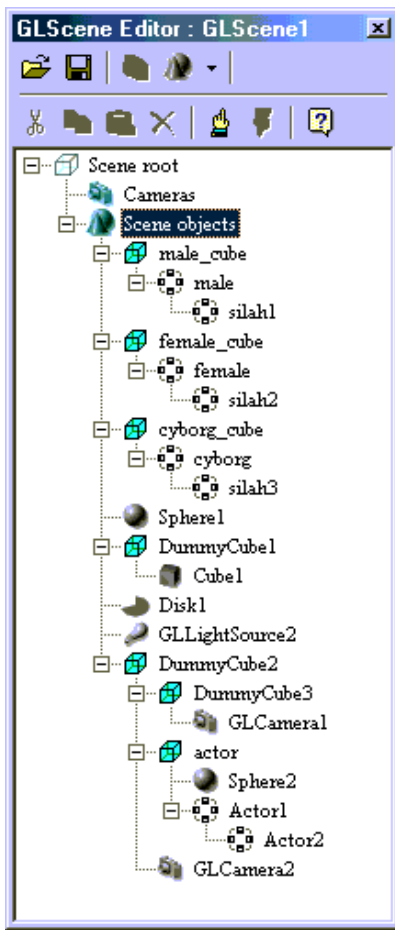
Programlama kısmı nerede olacak diye başlarda zorlanabiliriz, örneklerin yardımı ile yavaş yavaş çözüyoruz. Artık bildiğimiz form nesnelerini tamamen terkederek, menü, arabirim, ayarlar gibi kısımları yapabiliriz. Pencere görünümünden tam ekran görüntüye geçebiliriz.

Her iki proje de tam bitmedi. Fakat elimden geldiğince basit şeyler elde etmeye çalıştım. İlk örnekte, şimdilik Doom oyunu benzeri klayveden oklarla ileri geri, fare ile de büyütme (scroll – kaydırma tekerleği ile), çevirme işlemleri yapılabilir. F7 tepe kamerası, F8 ilk insan açısı, ESC kapatma, Shift ve ok tuşları hızlı ilerleme, nümerik klavyedeki sıfır tuşu ile eğilme, Enter ile ateş etme (?!), boşluk tuşu ile zıplama (ses de kullanıyoruz), 1 tuşu ile waste.md2, 2 tuşu ile erkek (cipher) tris.md2, 3 ile kadın (Athena) tris.md2 ve 4 ile de cyborg (oni911) tris.md2 yükleniyor. Ana menüden diğer aktörler de görülebilir. Ortadaki kutu gibi yere çarparak hareket edememe gibi duvarlara çarpınca içine girilmemesi sağlanabilir, ama titreme nasıl engellenebilir bilmiyorum. Ayrıca kavun içi renkli

küçük küreye çarpmayı başarabilerseniz patladığını görebilirsiniz. Ateş etmek için ışın (ThorFX) kullanmak da saçma gelebilir, daha basit bir çözüm bulamadım. Bir de klavye tuşları aynı anda kullanılamıyor, hem ateş hem eğilme gibi kombine hareketler yok. İkinci örnekte ise çözünürlük değiştirme, renk ayarları, form ve fare ayarları, dönen nesneler, ekranda zaman, FPS – frames per second (saniyedeki kare sayısı), çizgiler, daire çizmek, resim yüklemek gibi özellikler bulabilirsiniz. Türk yapımı oyun olan Pusu gibi bir şeyler de yapmak bizim için şimdilik biraz zor. Tek başımıza hikaye, harita tasarımı, ses efektleri, gibi şeyleri halletmek bizi aşıyor. Basit oyun yapmak veya bu örnekle dönem ödevlerinizde yardımcı olmak yeterli.

Her iki örnekteki kodları kitapta vermeyeceğim. Onun yerine form üzerindeki nesnelerden biraz bahsedeceğim. Elinizde CD olmadığını düşünürsek zaten kod yığınının hiçbir anlamı olmaz. İlk programımız olan ActorTwoCam örneğimiz iki formdan oluşuyor. İlki asıl pencere ve diğeri oyuncu seçme penceresi.

Temel nesneler GLSceneViewer1, GLScene1, GLCadencer1 ve CollisionManager1. Geri kalan GLThorFXManager1 ve GLFireFXManager1 nesneleri GLScene1 üzerindeki herhangi bir nesneye atanabilen efektlerdir. GLSoundLibrary1 nesnesi ile kütüphaneye eklediğimiz ses dosyalarını çalabilmemiz sağlanıyor. GLScene1 nesnesine çift tıklandığında sahnedeki tüm maddelere, ışıklara, kameralara ulaşabiliyoruz. GLSceneViewer1 ile çalışma anında karşılaşacağımız sahneyi hazırlıyoruz. GLCadencer1 ise zaman ayarlayıcısıdır. CollisionManager1 olmadan da iki nesnenin birbirine değip değmediğini anlayamayız. Kullanmadığımız diğer nesnelerin işe yaramadığı anlamı çıkmasın, AviRecorder, GLNavigator, BitmapFont gibi faydalı nesneleri de siz keşfedebilirsiniz.



Basit bir sahnede muhakkak GLSceneViewer1, GLScene1 ve GLCadencer1 nesnelerini bulundurmamız gerekir. Daha sonra da GLSceneViewer1 (Align:alClient yapsak daha iyi olur) ile çözünürlük, hangi kamera açısı olacak gibi ayarları yapabiliriz. Kamera GLScene1 çift tıklandığında araç çubuğundan eklenebilir. İki kameramız aşağıda belirtiliyor. Bu örnekteki sahne elemanları resimdeki gibidir:

Araç çubuğu basit. Ayrı bir dosya olarak kaydedebilme ve açma özelliği var. Kamera ve obje ekleme düğmesi, kes, kopyala, yapıştır, sil, yukarı hareket ettir, aşağı hareket ettir ve OpenGL özellikleri düğmeleri bulunuyor. Nesnelere fare ile sağ tıklayıp da aynı menülere ulaşabilirsiniz. Mavi küpler dikkatinizi çekmiştir. Bunlar gözükmezler fakat bizim için referans noktaları ve gruplar oluştururlar. DummyCube – Yapay Küp’ler olmadan nesnelerinizi yerleştirirseniz, uzayda kaybolup gidersiniz. Onlar sayesinde ayağımız yere(?) basıyor. Scene objects – sahne nesneleri ile GLCamera, GLLightSource, Sphere, Cube gibi nesneleri temel olarak kullanıyoruz.

İsterseniz yeni nesne dediğiniz menüdeki diğer işlevsel nesnelere de bakalım:

- ✓ Basic Geometry (basit geometrik şekiller) : Çizgi, poligon, küre gibi nesneler
- ✓ Advanced Geometry (ileri geometrik şekiller) : Çoklu poligon, Revolution Solid (dönerek elde edilen katı madde)
- ✓ Mesh Objects (bağ nesneleri) : Aktör, serbest yapı
- ✓ Procedural Objects (formüle bağlı nesneler) : partiküller, yeryüzü şekilleri
- ✓ Special Objects (özel objeler) : gökyüzü, boşlukta yazı, ok
- ✓ HUD Objects (Head up display – ön panel nesneleri) : İki boyutlu olan yazı gösterge bilgileri gibi işlevi olan nesneler
- ✓ Direct OpenGL : Doğrudan OpenGL render edilmesi
- ✓ Proxy Object : yetkili nesne, nesne kopyalamada kullanılabilir

Tek nesne içinde bu kadar özellik ilk kez görüyoruz. Alt elemanları tıklayıp özelliklerini karıştırabilirsiniz, zamanla bu ayarları çözebilirsiniz. GLScene nesnesini anladığınızda tüm programı avucunuzun içinde bulabilirsiniz. ChartFX nesnesi de güzel ama bu gerçekten daha yetenekli.

Nesnelerin sıraları önemli. Üstte ve altta olması önceliği belirtir. Alt katmana eklenmesi de grubun içindeki sırayı belirler.

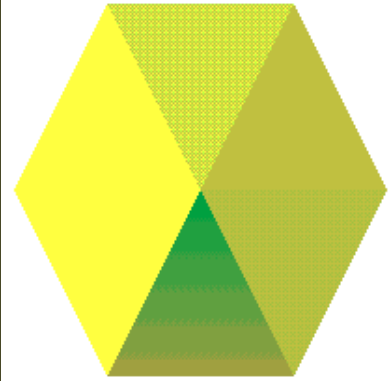
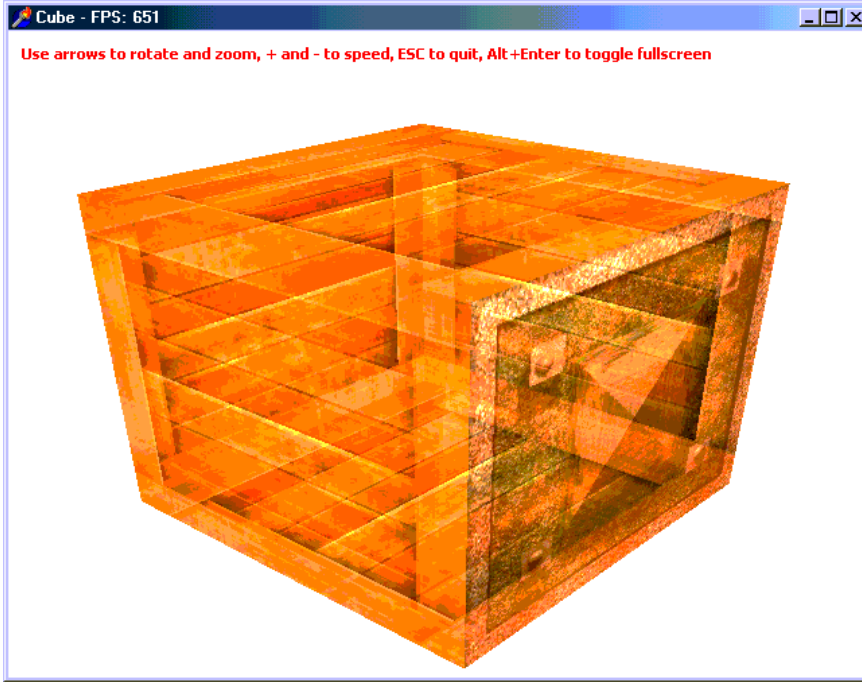
Daha x ve y boyutunu anlamadan, ekstra z boyutu bizi zorlayabilir. Bence burada nesnelerin koordinatları daha kolaylaştırılabilirdi. Çoğu zaman yerleştirdiğiniz nesneleri kaybedebiliyorsunuz. 3DSMax gibi programlara alışık olanlar fare ile nesneleri yerleştirmeyi daha çok sevebilirler. Rakamsal olarak x, y ve z ile uğraşmak biraz bıktırıyor. Her nesnenin benzer özellikleri var. Material (renk ve doku bilgileri), Position (sahte küp içindeki koordinat), Scale (ölçeklendirme), Up (koordinat bilgisi), Direction (koordinat bilgisi), Effects (efektler) gibi...

Başka bir boyut daha olan zaman olayları da TGLMovement, TGLMovementPath – yol ve TGLPathNode mantığı ile düzenleniyor. Mesela animasyon olarak bir uçağın düşmesi ve içinden canlı çıkan kahramanın sahnesi gibi düşünülebilir. GLScene içinde Pathcontrol adındaki bir örnekte belli bir yol üzerinde hareket eden küp bulabilirsiniz. Ayrıca diğer örneklerde gerçekten de çok hoş alev, partikül, ışın ve ışık efektleri bulabilirsiniz. Bir nesnenin Effects özelliğine ThorFX veya FireFX bağlamak kolay.

Programlama kısmı ise hala biraz toz duman... Çünkü yardım sistemi pek yardımcı olmuyor. FormCreate içine ilk açılışta GLActor1'in yüklenmesi, silahının yüklenip Actor1 ile senkron olması, yer şekillerinin dokularının yüklenmesi gibi olayları yazıyoruz. Eğer CD'den kodları incerseniz, bir çok yerde açıklama satırları halinde kodlar görebilirsiniz. Klavye tuş basımları ve fare tuşları ayarlandıktan sonra projemizi çalıştırıp denemelerde bulunabiliriz. GLCadencer1 nesnesinin tek olayı olan GLCadencer1Progress içine devamlı izlenen olaylar olan tuş basımı ve mayın küremizin kodlarını yazıyoruz. Artık bundan sonra, nesnelere Path - yol oluşturmayı öğrenmek, daha etkin kod yazmak, bizim hareketimizle tetiklenen diğer olayları yazmak, daha hoş ve gerçekçi materyaller elde etmek geliyor.

İkinci örnek platform (Mario benzeri) oyunları için daha uygun. Görsel olarak tasarlamak yerine herşeyi kodlarla yapmanız gerekli. Programlamayı sevenlere göre... DXTimer ve DXDraw nesneleri bizim için yeterli. Geri kalan nesneler resim, müzik kütüphaneleri, joystick ve network – ağ ayarları gibi işlemleri yaparlar. İsterseniz sadece işinizi gören ses kütüphanesini veya ağ nesnesini başka programlarda kullanabilirsiniz. Tercuman programımda bu DXWavList nesnesini kullanabilirdim. Programın bulunduğu dizindeki *.x uzantılı dosyaları 3DSMax veya Lightwave ile hazırlayabilirsiniz. Aslında *.3ds dosyalarını *.x formatına çeviren programlar da var. Bu dosyalarda Mesh Object – bileşik nesne olarak hazırlanmışlardır, yani sadece maddenin büküm, köşe gibi bilgileri saklanır.

D3DUtils.pas dosyasını incerseniz poligon, çizgi, vektörler, matrisler ile ilgili matematiksel kodlar bulabilirsiniz. DX2 klasöründeki örneğimizde, saydam bir sandığı klavye yön tuşları ile dönderdiğimi, ileri geri hareket ettirdiğimi görebilirsiniz. DXImageList kullanarak sandığın yüz dokularını yüklüyoruz.



55

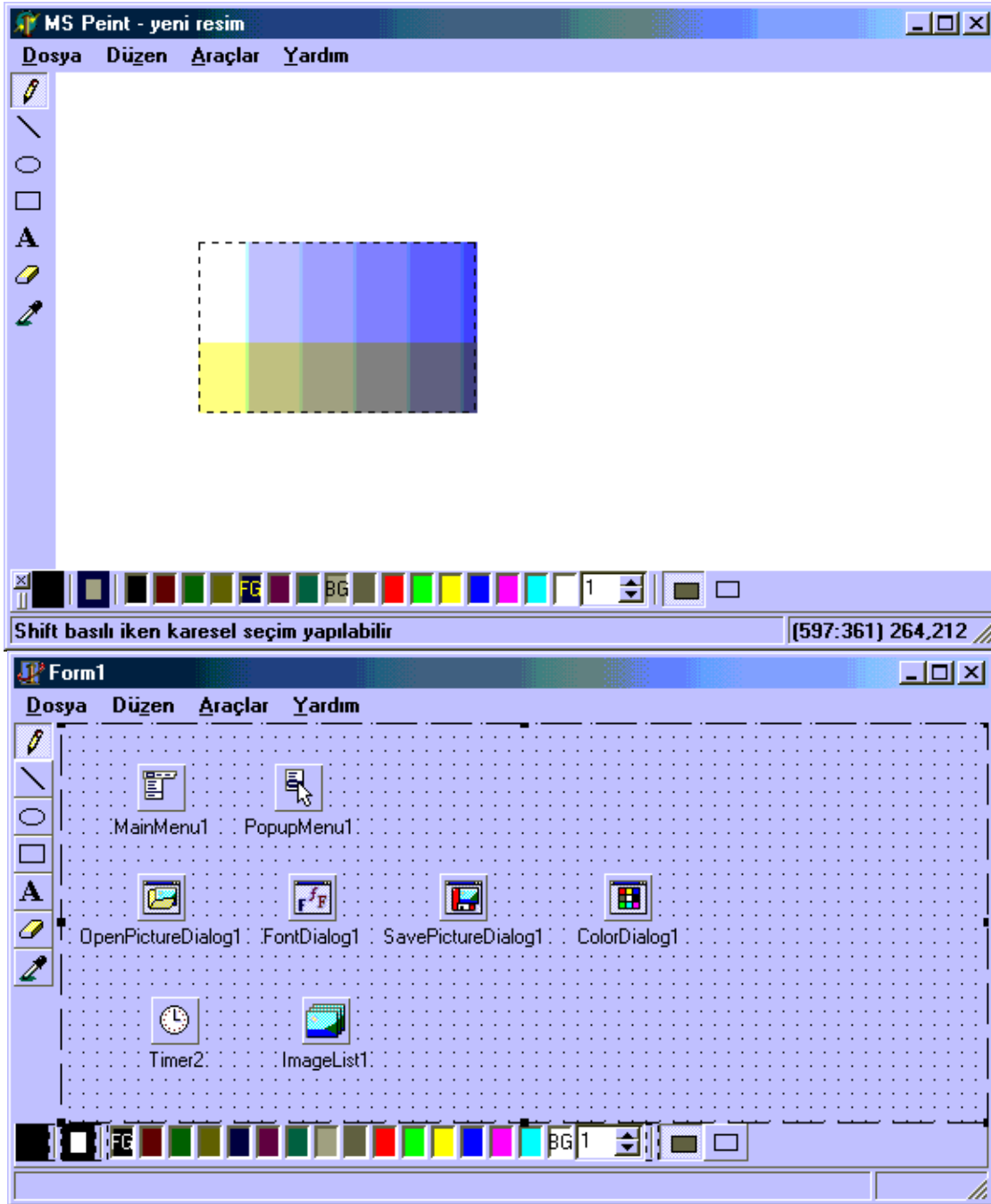
200

Resim 28 – DirectX örneği (DX2 dizindeki proje) ve tut2.exe (3Dfx çalışması)

20. MS Peint (PaintBrush yapımı)

(Ornekler\Delphi\Peint\MSPeint.dpr)

MSPaint veya diğer adı ile PaintBrush programına benzer bir program yazıyoruz. *.bmp dosya türünde dosya açma, kaydetme ve oluşturma işlemleri yapabiliyoruz. Ayrıca duvar kağıdı olarak ayarlamak, saydam yapıştırmak, renkleri ters çevirmek, renk doldurmak, rengi başka bir renk ile değiştirmek, büyüteç etkisi yapmak, boyutlandırmak gibi işlevler de kazandırabiliyoruz. Toolbar, MainMenu ve PopupMenu gibi nesneler de bize yardımcı oluyor. Delphi 5 ile hazırladığım bu örnekte Toolbar'ların sürüklenebildiği dikkatinizi çekebilir. Yerleştireceğiniz Panel'lerin koordinesi ile tekrar eski yerlerine almayı yapabilirsiniz. Araç çubuklarının Dock - gömülme özelliği Delphi'de kod yazmaya gerek kalmadan halledilebiliyor. MSPaint'deki gibi çevirme, uzatma, yazdırma, tarayıcı veya kameradan alma gibi özelliklerimiz yok. Dosya dönüştürme ve fırça gibi araçları siz de ekleyebilirsiniz. Geri al komutu yerinde yeller esiyor...



Resim 29 – Peint çalışmamız

Kodlar:

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, Menus, ToolWin, ComCtrls,
  ExtCtrls, ColorGrd, ExtDlgs, ImgList, StdCtrls, Spin, clipbrd;
type
  TForm1 = class(TForm)
    A1: TMenuItem;
    Aralar1: TMenuItem;
    Artr1: TMenuItem;
    Azalt1: TMenuItem;
    Boyutlandr1: TMenuItem;
    Byt1: TMenuItem;
    Byte1: TMenuItem;
    ColorDialog1: TColorDialog;
    ColorGrid1: TColorGrid;
    Dosya1: TMenuItem;
    DuvarKadOlarak1: TMenuItem;
    Dzen1: TMenuItem;
    FontDialog1: TFontDialog;
    Hakknda1: TMenuItem;
    Image1: TImage;
    ImageList1: TImageList;
    Kapat1: TMenuItem;
    Kaydet1: TMenuItem;
    Kes1: TMenuItem;
    Kes2: TMenuItem;
    Klt1: TMenuItem;
    Kopyala1: TMenuItem;
    Kopyala2: TMenuItem;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N4: TMenuItem;
    OpenPictureDialog1: TOpenPictureDialog;
    Panel1: TPanel;
    PopupMenu1: TPopupMenu;
    Renk1: TMenuItem;
    Renkler1: TMenuItem;
    SavePictureDialog1: TSavePictureDialog;
    SaydamYaptr1: TMenuItem;
    Seiliblgeyiiftrenkdoldur1: TMenuItem;
    SeiliRengiburenkiledeitir1: TMenuItem;
    SeiliRengiFGdeeriiledeitir1: TMenuItem;
    Shape1: TShape;
    Shape2: TShape;
    Siyahbeyaz1: TMenuItem;
    SpinEdit1: TSpinEdit;
    StatusBar1: TStatusBar;
    Timer2: TTimer;
    ToolBar1: TToolBar;
    ToolBar2: TToolBar;
    ToolButton1: TToolButton;
    ToolButton10: TToolButton;
    ToolButton11: TToolButton;
    ToolButton12: TToolButton;
    ToolButton2: TToolButton;
    ToolButton3: TToolButton;
    ToolButton4: TToolButton;
    ToolButton5: TToolButton;
    ToolButton6: TToolButton;
    ToolButton7: TToolButton;
```

```
    ToolButton8: TToolButton;
    ToolButton9: TToolButton;
    Yaptr1: TMenuItem;
    Yaptr2: TMenuItem;
    Yardm1: TMenuItem;
    Yaztipi1: TMenuItem;
    Yaztipi2: TMenuItem;
    Yeni1: TMenuItem;
    procedure Image1MouseMove(Sender: TObject; Shift:
    TShiftState; X, Y: Integer);
    procedure FormCreate(Sender: TObject);
    procedure Hakknda1Click(Sender: TObject);
    procedure Kapat1Click(Sender: TObject);
    procedure Yeni1Click(Sender: TObject);
    procedure A1Click(Sender: TObject);
    procedure Kaydet1Click(Sender: TObject);
    procedure Yaztipi1Click(Sender: TObject);
    procedure Dahafazlarenk1Click(Sender: TObject);
    procedure Kopyala1Click(Sender: TObject);
    procedure Kes1Click(Sender: TObject);
    procedure Yaptr1Click(Sender: TObject);
    procedure ToolButton1Click(Sender: TObject);
    procedure ToolButton2Click(Sender: TObject);
    procedure ToolButton3Click(Sender: TObject);
    procedure ToolButton4Click(Sender: TObject);
    procedure ToolButton5Click(Sender: TObject);
    procedure ToolButton6Click(Sender: TObject);
    procedure Image1MouseDown(Sender: TObject; Button:
    TMouseButton; Shift: TShiftState; X, Y: Integer);
    procedure ColorGrid1Change(Sender: TObject);
    procedure Image1MouseUp(Sender: TObject; Button:
    TMouseButton; Shift: TShiftState; X, Y: Integer);
    procedure Shape2ContextPopup(Sender: TObject;
    MousePos: TPoint; var Handled: Boolean);
    procedure Boyutlandr1Click(Sender: TObject);
    procedure SeiliRengiburenkiledeitir1Click(Sender:
    TObject);
    procedure Renkler1Click(Sender: TObject);
    procedure Siyahbeyaz1Click(Sender: TObject);
    procedure ToolButton11Click(Sender: TObject);
    procedure ToolButton12Click(Sender: TObject);
    procedure Artr1Click(Sender: TObject);
    procedure Azalt1Click(Sender: TObject);
    procedure Seiliblgeyiiftrenkdoldur1Click(Sender:
    TObject);
    procedure DuvarKadOlarak1Click(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
    procedure SaydamYaptr1Click(Sender: TObject);
    procedure Byt1Click(Sender: TObject);
    procedure Klt1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
  cizimtipi,basx,basy,thick:integer;
  renkal:TColor; kareSecim: TShape;
  gecici:TComponent; geciciResim:TImage;
  fname:string;
implementation
```



```
{ $R *.DFM }
```

```
procedure TForm1.Image1MouseMove(Sender: TObject;
  Shift: TShiftState; X,
  Y: Integer);
var renk:TColor;kalinlik:integer;
begin
  StatusBar1.Panels[1].text:='(' +inttostr(image1.Picture.Graphic
  c.Width)+': ' + inttostr (image1.Picture.Graphic.height)+')
  '+format('%d,%d',[x,y]);
  if (ssLeft in shift)and not(ssShift in shift) then begin
    image1.Canvas.pen.color:=shape2.Pen.Color;
    image1.Canvas.brush.color:=shape2.brush.Color;
    Image1.Canvas.pen.width:=SpinEdit1.Value;
    if cizimtipi=1 then Image1.Canvas.LineTo(x,y)
  else
    if cizimtipi=6 then begin
      renk:=Image1.Canvas.pen.color;
      kalinlik:=Image1.Canvas.pen.width;
      Image1.Canvas.pen.color:=clWhite;
      Image1.Canvas.pen.width:=SpinEdit1.Value;
    Image1.Canvas.Rectangle(x,y,x+SpinEdit1.Value,y+SpinEdit
    1.Value);
      Image1.Canvas.pen.color:=renk;
      Image1.Canvas.pen.width:=kalinlik;
    end;
  end;
end;
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  image1.Canvas.LineTo(0,0);
  image1.Canvas.brush.Style:=bssolid;
  cizimtipi:=1; thick:=5;
  Caption:='MS Peint - yeni resim';
  fname:='bos';
end;
```

```
procedure TForm1.Hakknda1Click(Sender: TObject);
begin
  MessageDlg('MS Peint 1.0'+chr(13)+'Kasım
  2001',mtCustom,[mbOk],0);
end;
```

```
procedure TForm1.Kapat1Click(Sender: TObject);
begin close; end;
```

```
procedure TForm1.Yeni1Click(Sender: TObject);
begin
  image1.Canvas.pen.Color:=clWhite;
  image1.Canvas.brush.Color:=clWhite;
  image1.Canvas.Rectangle(0,0,700,700);
  Caption:='MS Peint - yeni resim';
  fname:='bos';
  gecici:=Components[componentcount-1];
  //secimi tekrar aktif etmek için shape nesnesini siliyoruz
  if gecici is TShape then
    gecici.free;
end;
```

```
procedure TForm1.A1Click(Sender: TObject);
begin
  OpenPictureDialog1.Execute;
  if OpenPictureDialog1.FileName<>'' then begin
    image1.Picture.LoadFromFile (OpenPictureDialog1
    .FileName);
```

```
caption:='MS Peint - '+OpenPictureDialog1.FileName;
fname:=OpenPictureDialog1.FileName;
end;
gecici:=Components[componentcount-1];
//secimi tekrar aktif etmek için shape nesnesini siliyoruz
if gecici is TShape then
  gecici.free;
end;
```

```
procedure TForm1.Kaydet1Click(Sender: TObject);
begin
  SavePictureDialog1.Execute;
  if SavePictureDialog1.FileName<>'' then begin
    image1.Picture.SaveToFile(SavePictureDialog1.FileName);
    caption:='MS Peint - '+SavePictureDialog1.FileName;
    fname:=savePictureDialog1.FileName;
  end;
  gecici:=Components[componentcount-1];
  if gecici is TShape then
    gecici.free;
end;
```

```
procedure TForm1.Yaztipi1Click(Sender: TObject);
begin
  FontDialog1.Font:=image1.Canvas.Font;
  FontDialog1.Execute;
  image1.Canvas.Font:=FontDialog1.Font;
end;
```

```
procedure TForm1.Dahafazlarenk1Click(Sender: TObject);
begin
  ColorDialog1.Color:=Image1.Canvas.pen.Color;
  if ColorDialog1.Execute then
    Image1.Canvas.pen.Color:=ColorDialog1.Color;
end;
```

```
procedure TForm1.Kopyala1Click(Sender: TObject);
var i,j:integer;
begin
  gecici:=Components[componentcount-1];
  if gecici is TShape then begin
    geciciResim := Timage.Create (self);
    geciciResim.Parent:= self;
    geciciResim.Visible:=false;
    geciciResim.Width:=kareSecim.Width;
    geciciResim.height:=kareSecim.height;

    for i:=0 to kareSecim.Width do
      for j:=0 to kareSecim.height do
        geciciResim.Canvas.Pixels[i,j]:=image1.Canvas.Pixels
          [i+karesecim.left-toolbar1.Width,j+karesecim.top];
        Clipboard.Assign(geciciResim.Picture);
        geciciResim.free;
      // gecici:=Components[componentcount-1];
      // if gecici is TShape then gecici.free;
      kareSecim.Visible:=false;
    end else
      Clipboard.Assign(Image1.Picture);
  end;
```

```
procedure TForm1.Kes1Click(Sender: TObject);
var i,j:integer;
begin
  Kopyala1Click(Sender); //önce kopyala sonra sil...
  gecici:=Components[componentcount-1];
  if gecici is TShape then begin
```

```

for i:=0 to kareSecim.Width do
  for j:=0 to kareSecim.height do
    image1.Canvas.Pixels[i+karesecim.left-
toolbar1.Width,j+karesecim.top]:=clWhite;
  end
  else begin
    image1.Canvas.pen.Color:=clWhite;
    image1.Canvas.brush.Color:=clWhite;
    image1.Canvas.Rectangle(0,0,image1.Width,image1.height);
  end;
end;

```

```

procedure TForm1.Yaptr1Click(Sender: TObject);
var
  Bitmap: TBitmap;
begin
  if Clipboard.HasFormat(CF_BITMAP) then
  begin
    Bitmap := TBitmap.Create;
    try
      Bitmap.Assign(Clipboard);
      Image1.Canvas.Draw(0, 0, Bitmap);
    finally
      Bitmap.Free;
    end;
  end;
end;

```

```

procedure TForm1.ToolButton1Click(Sender: TObject);
begin cizimtipi:=1; end;

```

```

procedure TForm1.ToolButton2Click(Sender: TObject);
begin cizimtipi:=2; end;

```

```

procedure TForm1.ToolButton3Click(Sender: TObject);
begin cizimtipi:=3; end;

```

```

procedure TForm1.ToolButton4Click(Sender: TObject);
begin cizimtipi:=4; end;

```

```

procedure TForm1.ToolButton5Click(Sender: TObject);
begin cizimtipi:=5; end;

```

```

procedure TForm1.ToolButton6Click(Sender: TObject);
begin cizimtipi:=6; end;

```

```

procedure TForm1.Image1MouseDown(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var gecici:TComponent;
begin
  basx:=x;
  basy:=y;
  Image1.Canvas.MoveTo(x,y);
  if ToolButton7.Down then
  begin
    renkal:=Image1.Canvas.Pixels[x,y];
    shape1.Brush.Color:=renkal;
    shape1.pen.Color:=renkal;
  end;

```

```

  gecici:=Components[componentcount-1];
  //secimi tekrar aktif etmek için shape nesnesini siliyoruz
  if gecici is TShape then
    gecici.free;
end;

```

```

procedure TForm1.ColorGrid1Change(Sender: TObject);
begin
  Shape2.Pen.Color:=ColorGrid1.ForegroundColor;
  Shape2.Brush.Color:=ColorGrid1.backgroundColor;
end;

```

```

procedure TForm1.Image1MouseUp(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var metin:string;
begin
  image1.Canvas.pen.color:=shape2.Pen.Color;
  image1.Canvas.brush.color:=shape2.brush.Color;
  if ToolButton11.Down then
    image1.Canvas.brush.Style:=bsSolid;
  if ToolButton12.Down then
    image1.Canvas.brush.Style:=bsClear;
  image1.Canvas.pen.width:=SpinEdit1.Value;
  if ssShift in shift then //shift ile seçim (kare)
  begin
    kareSecim := Tshape.Create (self);
    kareSecim.Parent:= self;
    kareSecim.ShowHint:=true;
    kareSecim.Hint:='Menüden kes, kopyala gibi işlemleri
yapabilirsiniz';
    kareSecim.Pen.Style:=psDot;
    kareSecim.Brush.Style:=bsClear;
    if x-basx<0 then
    begin
      kareSecim.Width:=basx-x;
      kareSecim.Left:= x+toolbar1.Width;
    end else begin
      kareSecim.Width:=x-basx;
      kareSecim.Left:= basx+toolbar1.Width;
    end;
    if y-basy<0 then
    begin
      kareSecim.Height:=basy-y;
      kareSecim.Top:= y;
    end else begin
      kareSecim.Height:=y-basy;
      kareSecim.Top:= basy;
    end;
  end else begin
    if cizimtipi=2 then image1.Canvas.LineTo(x,y);
    if cizimtipi=3 then image1.Canvas.Ellipse(basx,basy,x,y);
    if cizimtipi=4 then image1.Canvas.Rectangle(basx,basy,x,y);
    if cizimtipi=5 then
    begin
      InputQuery('MS Paint','Metin girişi',metin);
      image1.Canvas.TextOut(x,y,metin);
    end;
  end;
end;

```

```

procedure TForm1.Shape2ContextPopup(Sender: TObject;
MousePos: TPoint;
var Handled: Boolean);
begin
  ColorDialog1.Color:=Image1.Canvas.pen.Color;
  if ColorDialog1.Execute then
    shape2.Pen.Color:=ColorDialog1.Color;

  ColorDialog1.Color:=Image1.Canvas.Brush.Color;
  if ColorDialog1.Execute then

```

```

    shape2.brush.Color:=ColorDialog1.Color;
end;

procedure TForm1.Boyutlandr1Click(Sender: TObject);
var
    genis,yuksek:string;
begin
    InputQuery('Genişlik','Eski
değeri:'+inttostr(Image1.Picture.Graphic.Width)+'', Boş
geçebilirsiniz',genis);
    InputQuery('Yükseklik','Eski
değeri:'+inttostr(Image1.Picture.Graphic.height)+'', Boş
geçebilirsiniz',yuksek);
    if genis<>'' then Image1.Picture.Graphic.Width :=
strtoint(genis);
    if yuksek<>'' then Image1.Picture.Graphic.Height:=
strtoint(yuksek);
end;

procedure TForm1.SeiliRengiburenkiledeitir1Click(Sender:
TObject);
var i,j:integer;buradakiRenk:Tcolor;
begin
    buradakiRenk:=ColorGrid1.ForegroundColor;
    for i:=0 to Image1.Picture.Graphic.Width do
    for j:=0 to Image1.Picture.Graphic.height do
    begin
        if image1.Canvas.Pixels[i,j]=renkal then
            image1.Canvas.Pixels[i,j]:=buradakiRenk;
        end;
    end;
end;

procedure TForm1.Renkler1Click(Sender: TObject);
var i,j:integer;
begin
    for i:=0 to Image1.Picture.Graphic.Width do
    for j:=0 to Image1.Picture.Graphic.height do
    begin
        image1.Canvas.Pixels[i,j]:=$02FFFFFF-
image1.Canvas.Pixels[i,j];
    end;
end;

procedure TForm1.Siyahbeyaz1Click(Sender: TObject);
var i,j:integer;
begin
    for i:=0 to Image1.Picture.Graphic.Width do
    for j:=0 to Image1.Picture.Graphic.height do
    begin
        if image1.Canvas.Pixels[i,j]>($FFFFFF/2) then
            image1.Canvas.Pixels[i,j]:=clwhite else
            image1.Canvas.Pixels[i,j]:=clBlack;
        end;
    end;
end;

procedure TForm1.ToolButton11Click(Sender: TObject);
begin
    image1.Canvas.brush.Style:=bsclear;
end;

procedure TForm1.ToolButton12Click(Sender: TObject);
begin
    image1.Canvas.brush.Style:=bsSolid;
end;

procedure TForm1.Artr1Click(Sender: TObject);

```

```

//renk artırmak spinedit'e bağlı
var i,j:integer;
    r,g,b:byte;
begin
    for i:=0 to Image1.Picture.Graphic.Width do
    for j:=0 to Image1.Picture.Graphic.height do
    begin
        r:=getrvalue(image1.Canvas.Pixels[i,j])-SpinEdit1.Value;
        g:=getgvalue(image1.Canvas.Pixels[i,j])-SpinEdit1.Value;
        b:=getbvalue(image1.Canvas.Pixels[i,j])-SpinEdit1.Value;
        image1.Canvas.Pixels[i,j]:=RGB(r,g,b);
    end;
end;

procedure TForm1.Azalt1Click(Sender: TObject);
var i,j:integer;
    r,g,b:byte;
begin
    for i:=0 to Image1.Picture.Graphic.Width do
    for j:=0 to Image1.Picture.Graphic.height do
    begin
        r:=getrvalue(image1.Canvas.Pixels[i,j])+SpinEdit1.Value;
        g:=getgvalue(image1.Canvas.Pixels[i,j])+SpinEdit1.Value;
        b:=getbvalue(image1.Canvas.Pixels[i,j])+SpinEdit1.Value;
        image1.Canvas.Pixels[i,j]:=RGB(r,g,b);
    end;
end;

procedure TForm1.Seiliblgeyiiftrenkdoldur1Click(Sender:
TObject);
var i,j,x1,y1:integer;renk1,renk2:TColor;artim:longint;
    r,g,b:byte;
begin
    gecici:=Components[componentcount-1];
    if gecici is TShape then
    begin
        x1:=kareSecim.Left;
        y1:=kareSecim.Top;
        for i:=0 to kareSecim.Width do
        for j:=0 to kareSecim.height do
        begin
            r:=getrvalue(image1.Canvas.Pixels[i+x1-
toolbar1.Width,j+y1])-i;
            g:=getgvalue(image1.Canvas.Pixels[i+x1-
toolbar1.Width,j+y1])-i;
            b:=getbvalue(image1.Canvas.Pixels[i+x1-
toolbar1.Width,j+y1])-j;
            image1.Canvas.Pixels[i+x1-
toolbar1.Width,j+y1]:=RGB(r,g,b);
        end;
    end;
end;

procedure TForm1.DuvarKadOlarak1Click(Sender: TObject);
begin
    if fname<>'bos' then
        SystemParametersInfoA(SPI_SETDESKWALLPAPER, 0,
PChar(FName), SPIF_UPDATEINIFILE);
end;

procedure TForm1.Timer2Timer(Sender: TObject);
begin
    StatusBar1.Panels[0].Text:=Application.Hint;
end;

procedure TForm1.SaydamYaptr1Click(Sender: TObject);

```

```

var
  bmpXOR, bmpAND, bmpINVAND, bmpTarget: TBitmap;
oldcol: Longint; s:Tbitmap; trcol, clr: Tcolor;
begin
  if Clipboard.HasFormat(CF_BITMAP) then
    begin
      s := TBitmap.Create;
      s.Assign(Clipboard);
      trcol:=clwhite; //saydam renk
    try
      bmpAND := TBitmap.Create;
      bmpAND.Width := s.Width;
      bmpAND.Height := s.Height;
      bmpAND.Monochrome := True;
      oldcol := SetBkColor(s.Canvas.Handle,
ColorToRGB(TrCol));
      BitBlt(bmpAND.Canvas.Handle, 0,0,s.Width,s.Height,
s.Canvas.Handle, 0,0, SRCCOPY);
      SetBkColor(s.Canvas.Handle, oldcol);
      bmpINVAND := TBitmap.Create;
      bmpINVAND.Width := s.Width;
      bmpINVAND.Height := s.Height;
      bmpINVAND.Monochrome := True;
      BitBlt(bmpINVAND.Canvas.Handle, 0,0,s.Width,s.Height,
bmpAND.Canvas.Handle, 0,0, NOTSRCCOPY);
      bmpXOR := TBitmap.Create;
      bmpXOR.Width := s.Width;
      bmpXOR.Height := s.Height;
      BitBlt(bmpXOR.Canvas.Handle, 0,0,s.Width,s.Height,
s.Canvas.Handle, 0,0, SRCCOPY);
      BitBlt(bmpXOR.Canvas.Handle, 0,0,s.Width,s.Height,
bmpINVAND.Canvas.Handle, 0,0, SRCAND);
      bmpTarget := TBitmap.Create;
      bmpTarget.Width := s.Width;
      bmpTarget.Height := s.Height;
      BitBlt(bmpTarget.Canvas.Handle, 0,0,s.Width,s.Height,
image1.Canvas.Handle, 1,1, SRCCOPY);
      BitBlt(bmpTarget.Canvas.Handle, 0,0,s.Width,s.Height,
bmpAND.Canvas.Handle, 0,0, SRCAND);
      BitBlt(bmpTarget.Canvas.Handle, 0,0,s.Width,s.Height,
bmpXOR.Canvas.Handle, 0,0, SRCINVERT);
      BitBlt(image1.Canvas.Handle, 1,1,s.Width,s.Height,
bmpTarget.Canvas.Handle, 0,0, SRCCOPY);
    finally
      bmpXOR.Free;
      bmpAND.Free;
      bmpINVAND.Free;
      bmpTarget.Free;
    end;{End of TRY section}
  image1.Invalidate;
  image1.repaint;
end;
end;

```

```

procedure TForm1.Byt1Click(Sender: TObject);
var
  Srect,Drect,PosForme:TRect;
  iWidth,iHeight,DmX,DmY:Integer;
  iTmpX,iTmpY:Real;
  C:TCanvas;

```

```

ZoomFactor: integer;
Kursor:TPoint;
begin
  GetCursorPos(Kursor);
  iWidth:=Image1.Width;
  iHeight:=Image1.Height;
  Drect:=Rect(0,0,iWidth,iHeight);
  { 2x kadar büyütür. 4x için 2, ...}
  ZoomFactor := 1;
  iTmpX:=iWidth / (ZoomFactor * 4);
  iTmpY:=iHeight / (ZoomFactor * 4);
  Srect:=Rect(Kursor.x,Kursor.y,Kursor.x,Kursor.y);
  InflateRect(Srect,Round(iTmpX),Round(iTmpY));
  If Srect.Left<0 then OffsetRect(Srect,-Srect.Left,0);
  If Srect.Top<0 then OffsetRect(Srect,0,-Srect.Top);
  If Srect.Right>(Screen.Width) then OffsetRect(Srect,-
(Srect.Right-Screen.Width),0);
  If Srect.Bottom>(Screen.Height) then OffsetRect(Srect,0,-
(Srect.Bottom-Screen.Height));
  C:=TCanvas.Create;
  try
    C.Handle:=GetDC(GetDesktopWindow);
    Image1.Canvas.CopyRect(Drect,C,Srect);
  finally
    C.Free;
  end;
end;

```

```

procedure TForm1.Klt1Click(Sender: TObject);
var
  Srect,Drect,PosForme:TRect;
  iWidth,iHeight,DmX,DmY:Integer;
  iTmpX,iTmpY:Real;  C:TCanvas;
  ZoomFactor: Integer;  Kursor:TPoint;
begin
  GetCursorPos(Kursor);
  iWidth:=Image1.Width;
  iHeight:=Image1.Height;
  Drect:=Rect(0,0,iWidth,iHeight);
  ZoomFactor := 1;
  iTmpX:=iWidth / 1;
  iTmpY:=iHeight / 1;
  Srect:=Rect(Kursor.x,Kursor.y,Kursor.x,Kursor.y);
  InflateRect(Srect,Round(iTmpX),Round(iTmpY));
  If Srect.Left<0 then OffsetRect(Srect,-Srect.Left,0);
  If Srect.Top<0 then OffsetRect(Srect,0,-Srect.Top);
  If Srect.Right>(Screen.Width) then OffsetRect(Srect,-
(Srect.Right-Screen.Width),0);
  If Srect.Bottom>(Screen.Height) then OffsetRect(Srect,0,-
(Srect.Bottom-Screen.Height));
  C:=TCanvas.Create;
  try
    C.Handle:=GetDC(GetDesktopWindow);
    Image1.Canvas.CopyRect(Drect,C,Srect);
  finally
    C.Free;
  end;
end;

end.

```

21. Sayı Bulmaca Oyunu (4 haneli sayı)

(Ornekler\Delphi\SayıBul2\Project1.dpr)

Öğrenci iken oynadığımız sayı bulma oyununun, bilgisayara karşı tek kişilik oynananı. 4 haneli tutan rakam olursa "+" artar, yeri hatalı bulunan rakamda da "-" artar. Sonuçta tüm haneler tutturulunca oyun bitiyor. Eski yazdığınız rakamı yandaki ListBox1 nesnesinde görebilirsiniz. En az sayıda bilmek önemli, süre tutulmasına gerek yoktur.

Tekrar rakam kullanılabilmesi de oyunun diğer seviyesi oluyor. Biraz daha zorlaştırıp, basamak sayısı da kullanıcıya bırakılabilirdi. Maskedit1 nesnesinin EditMask özelliğine **!9999;1;+** yazabiliriz. Diğer nesnelerin özel bir durumu yok.



Resim 30 – Sayı Bulmaca

Kodlar:

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, Mask, Buttons;

type
  TForm1 = class(TForm)
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    CheckBox1: TCheckBox;
    ListBox1: TListBox;
    MaskEdit1: TMaskEdit;
    StaticText1: TStaticText;
    StaticText2: TStaticText;
    StaticText3: TStaticText;
    StaticText4: TStaticText;
    procedure BitBtn2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure MaskEdit1KeyDown(Sender: TObject; var Key:
Word;
      Shift: TShiftState);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  deneme: integer;
  dizi:array [1..4] of integer;
```

```
dizi2:array [0..9] of integer;
sayac,kere: integer;
implementation
```

```
{ $R *.dfm }
```

```
procedure tusdoldur; //artık bu kod her yerde var
var a:integer;
begin
  if form1.CheckBox1.Checked then
    repeat
      a:=random(10);
      if dizi[sayac]=-1 then
        begin
          dizi[sayac]:=a;
          inc(sayac);
        end
      else tusdoldur;
    until sayac=5
  else
    repeat
      a:=random(10);
      if dizi2[a]=-1 then
        begin
          dizi[sayac]:=a;
          dizi2[a]:=a;
          inc(sayac);
        end
      else tusdoldur;
    until sayac=5;
end;
```

```
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
  close //kapat
end;
```

```

procedure TForm1.FormCreate(Sender: TObject);
var i:integer;
begin
randomize;
deneme:=0;
kere:=1;
for i:=1 to 4 do dizi[i]:=-1;
for i:=0 to 9 do dizi2[i]:=-1;
sayac:=1;
tusdoldur;
end;

```

```

procedure TForm1.BitBtn1Click(Sender: TObject);
var kullanıcı:string;           //Tamam düğmesi
    bilinen, tutulanyer:byte;
begin
kullanıcı:=MaskEdit1.Text;
bilinen:=0;
tutulanyer:=0;

```

```

if (kullanıcı[1]=' ')or(kullanıcı[2]=' ')or
(kullanıcı[3]=' ')or(kullanıcı[4]=' ') then
begin
    StaticText2.Caption:='Hatalı Giriş';
    exit;
end;
if ((kullanıcı[1]=kullanıcı[2])or
(kullanıcı[1]=kullanıcı[3])or
(kullanıcı[1]=kullanıcı[4])or
(kullanıcı[2]=kullanıcı[3])or
(kullanıcı[2]=kullanıcı[4])or
(kullanıcı[3]=kullanıcı[4]))
and (form1.CheckBox1.Checked=false) then begin
    StaticText2.Caption:='Hatalı Giriş';
    exit;
end;

```

```

staticText4.Caption:=inttostr(kere);
inc(kere);
StaticText2.Caption:='Kere';
if kullanıcı[1]=inttostr(dizi[1]) then inc(tutulanyer);
if kullanıcı[2]=inttostr(dizi[2]) then inc(tutulanyer);
if kullanıcı[3]=inttostr(dizi[3]) then inc(tutulanyer);
if kullanıcı[4]=inttostr(dizi[4]) then inc(tutulanyer);
if CheckBox1.Checked = false then begin

```

```

if (strtoint(kullanıcı[1]) in [dizi[2],dizi[3],dizi[4]]) then
inc(bilinen);
if (strtoint(kullanıcı[2]) in [dizi[1],dizi[3],dizi[4]]) then
inc(bilinen);
if (strtoint(kullanıcı[3]) in [dizi[1],dizi[2],dizi[4]]) then
inc(bilinen);
if (strtoint(kullanıcı[4]) in [dizi[1],dizi[2],dizi[3]]) then
inc(bilinen);
StaticText3.Caption:=inttostr(bilinen)+' - ;
'+inttostr(tutulanyer)+' + ';
end
else
StaticText3.Caption:=inttostr(tutulanyer)+' + ';
ListBox1.Items.Add(maskedit1.text+ ' '+inttostr(bilinen)+' - ;
'+inttostr(tutulanyer)+' + ' ');
if tutulanyer=4 then begin
MaskEdit1.Enabled:=false;
StaticText2.Caption:='KAZANDIN';
end;
end;

```

```

procedure TForm1.BitBtn3Click(Sender: TObject);
var i:integer;           //Yeni oyun
begin
deneme:=0;
kere:=1;
ListBox1.Clear;
MaskEdit1.Enabled:=true;
for i:=1 to 4 do dizi[i]:=-1;
for i:=0 to 9 do dizi2[i]:=-1;
sayac:=1;
tusdoldur;
MaskEdit1.Text:='';
StaticText4.Caption:='0';
StaticText2.Caption:='Kere';
StaticText3.Caption:='0 - ; 0 + ';
end;

```

```

procedure TForm1.MaskEdit1KeyDown(Sender: TObject; var
Key: Word; Shift: TShiftState);
begin
if key=VK_RETURN then BitBtn1.OnClick(self);
end;

end.

```

22. Veritabanı Grafik Analizi (DBChart kullanımı)

(Ornekler\Delphi\grafik_Chart\DB\Project1.dpr)

Delphi nesnelerinde gözetinca Chart, VTChart, DBChart ve ChartFX gibi birçok grafik nesnesi ile karşılaşırız. Sonuçta her türlü uygulamalarda basit de olsa bir veri dizisinin grafiğini kullanmamız gerekebilir. Sabit veri girişlerinin grafiğinin çıkarılması için oluşturduğumuz "Ornekler\Delphi\grafik_Chart\Project1.dpr" örneğini inceleyebilirsiniz. Şimdiki örneğin verileri ise database yani veritabanından geliyor.

veri.db dosyamızın genel yapısı:

Alan Adı	Tipi	Büyükklüğü
AdiSoyadi	A	32
Yasi	N	
Cinsiyeti	A	5
EgitimDurumu	A	32
Meslek	A	32
IlgiAlani	A	32

Tabii ki cinsiyeti Bay/Bayan, eğitim durumunu Ortaokul, Lise, Yüksekokul gibi bir listeden alması kullanıcının veri girişini daha kolaylaştırıyor. Bunun için DBGrid1 nesnesine çift tıklayarak, çıkan küçük pencereye sağ tıklayıp "Add All Fields" komutunu vererek istediğimiz alanın PickList özelliğini dolduruyoruz.

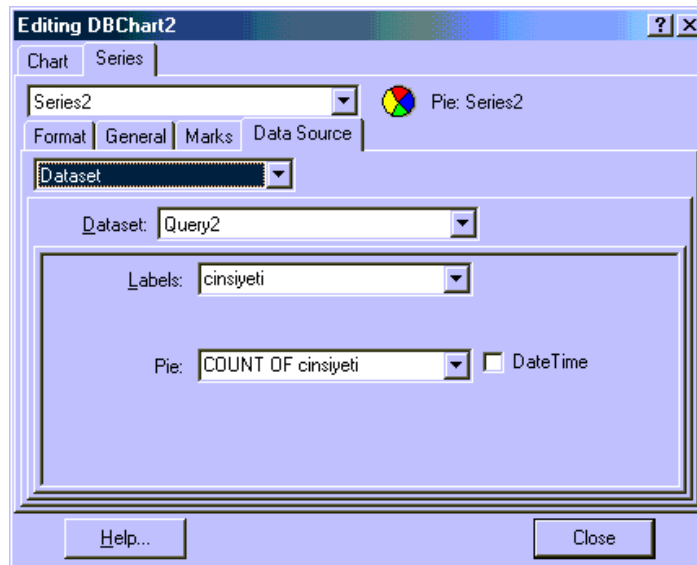
Grafiğin temel fikri, bir tür veride yüzde kaç giriş yapılmış olduğunu Pie – pasta grafik şeklinde görmektir. Diğer grafik tiplerine göre Bar – çubuk ve Pie – pasta daha anlaşılır. Bir grafiğin özelliklerine erişmek için ona çift tıklayınız.

DBChart1 nesnesini Cinsiyeti için kullanırsak şöyle yapabiliriz. Form üzerine attığımız Query1 nesnesinin SQL özelliğine:

```
select cinsiyeti, count (cinsiyeti)
from veri
group by cinsiyeti
```

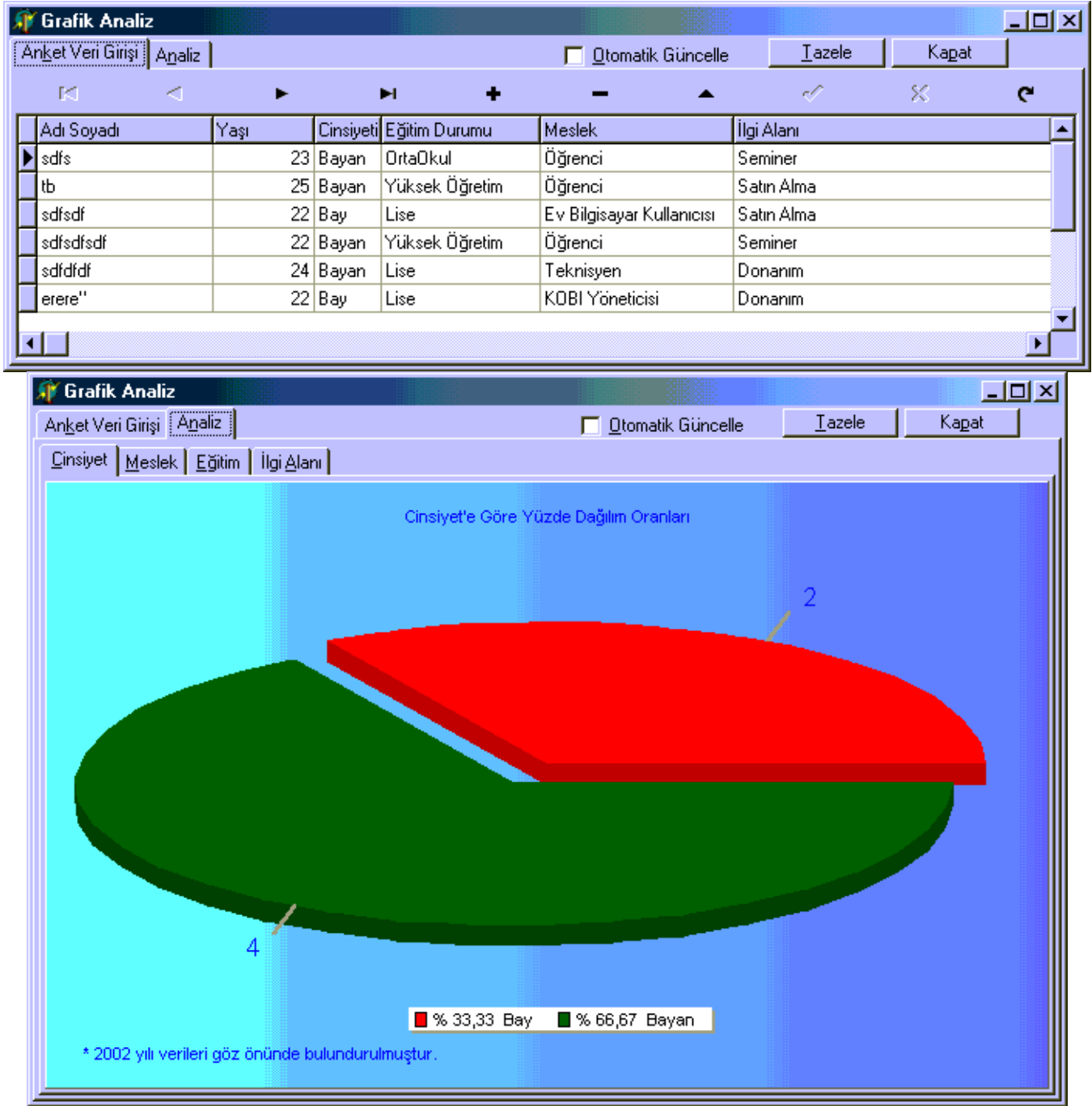
yazıp, Active özelliğini True yapıyoruz. Bu deyim ile veri.db'den bir alanı alıp, o alandan kaç adet olduğunu bularak süzmemizi sağlıyoruz. Örneğin Bay 3 adet, Bayan 4 adet gibi...

DBChart1, adından belli olduğu gibi verilerini bir tablo veya sorgudan alıyor. Üzerine çift tıklayıp, Chart sayfasında Add düğmesine basıyoruz. Eklediğimiz seriye Series sayfasından Datasource iç sayfasına tıklayıp, Dataset, Query1, cinsiyeti, COUNT OF cinsiyeti sırası ile giriş yapıyoruz. Bunun için aşağıdaki şekile bakınız.



Resim 31a – DBChart1'in seri bilgisi

Bundan sonra da yine resimlerdeki gibi bir grafik elde etmek isterseniz her türlü ayar bu şekilde saklı. Form tasarımı zamanımızı alırken, neredeyse hiç program yazmadan işimiz bitiyor. Sadece Timer1 ile veya Button2 ile güncellemek için gerekli birkaç satır komut yazıyoruz.



Resim 31b – Form tasarımı

Kodlar:

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes,  
Graphics, Controls, Forms,
```

```
Dialogs, ComCtrls, ExtCtrls, DBTables, DB, DBCtrls,  
Grids, DBGrids,  
StdCtrls, TeEngine, Series, TeeProcs, Chart, DbChart;
```

```
type
```

```
TForm1 = class(TForm)  
Button1: TButton;
```



```

Button2: TButton;
CheckBox1: TCheckBox;
DataSource1: TDataSource;
DBChart1: TDBChart;
DBChart2: TDBChart;
DBChart3: TDBChart;
DBChart4: TDBChart;
DBGrid1: TDBGrid;
DBNavigator1: TDBNavigator;
PageControl1: TPageControl;
PageControl2: TPageControl;
PieSeries1: TPieSeries;
Query1: TQuery;
Query2: TQuery;
Query3: TQuery;
Query4: TQuery;
Series1: TPieSeries;
Series2: TPieSeries;
Series3: TPieSeries;
Table1: TTable;
TabSheet1: TTabSheet;
TabSheet2: TTabSheet;
TabSheet3: TTabSheet;
TabSheet4: TTabSheet;
TabSheet5: TTabSheet;
TabSheet6: TTabSheet;
Timer1: TTimer;
procedure Button1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;

```

implementation

```

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
close
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  Button2Click(self);
  //böylece iki kat program yazmaktan kurtulduk
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  Query1.close; //biraz aptalca ve yavaş...
  Query1.Open;
  Query2.close;
  Query2.Open;
  Query3.close;
  Query3.Open;
  Query4.close;
  Query4.Open;
  DBChart1.RefreshData;
  DBChart2.RefreshData;
  DBChart3.RefreshData;
  DBChart4.RefreshData;
end;

procedure TForm1.CheckBox1Click(Sender: TObject);
begin
  timer1.Enabled:=CheckBox1.Checked;
end;

end.

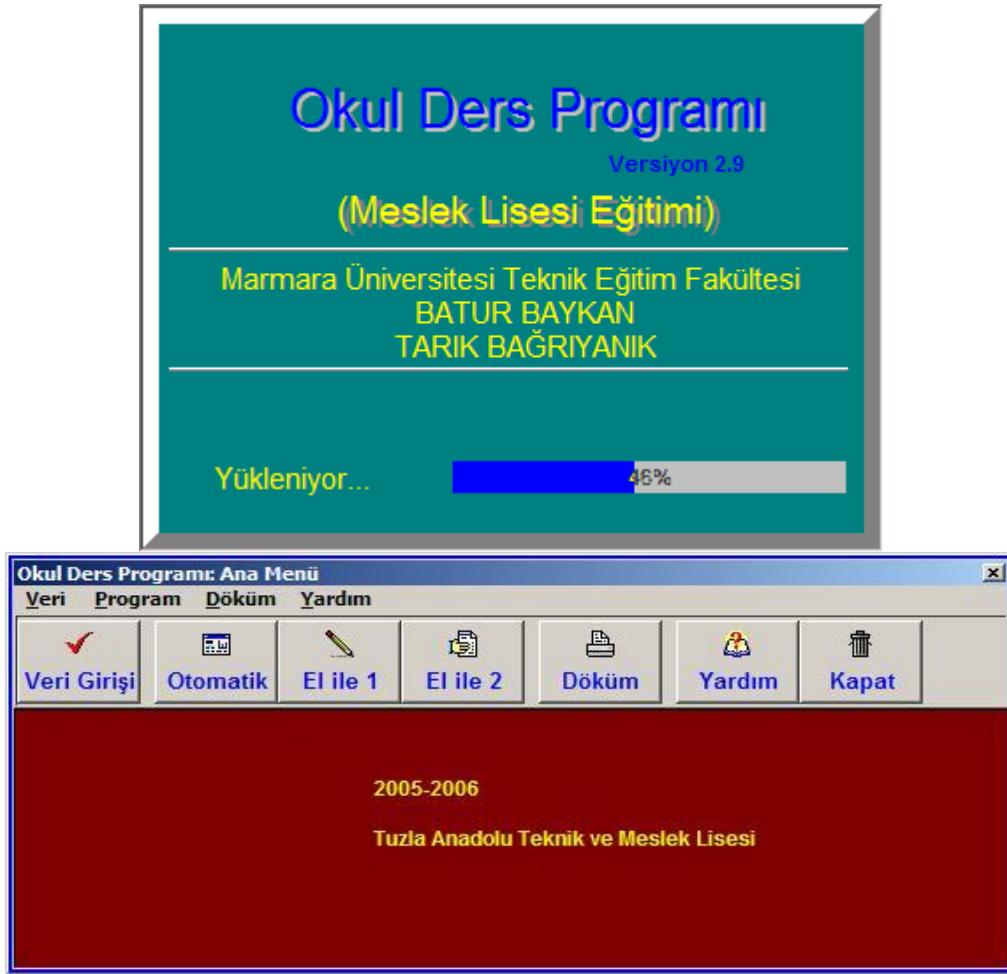
```

23. Okul Ders Programı (Haftalık ders programı yapımı)

(Ornekler\Delphi\DersProg.zip ve orgun.zip)

1998 yılındaki üniversite son sınıf tezimiz. Bu programın amacı Okul Ders Programı yapmaktır. Okul Ders Programı bir okulun ders kayıtlarının tutulabileceği ve ders programlarının hazırlanabileceği bir programdır. Bu program ile şube, öğretim elemanı (öğretmen) ve dershanelerin haftalık ders programı kayıtları tutulabilir ve bunların dökümleri alınabilir.

Batur Baykan arkadaşım ile yaptık, aradan uzun bir zaman geçtiği için artık kitaba da ekleyebilirim diye düşündüm. Yaklaşık 850 saatlik çalışma, 11000 satır olan kodlarını CD'den bakmanız gerekli, burada sadece bazı ekran görüntüleri ve algoritmasını ekledim. Delphi 7 ile tasarladığımız için son sürüm QuickRep yüklü Delphi'de rahatlıkla açılıp değişiklikler yapılabilir.



Resim 32a– Ders Programı açılış penceresi

Ders Programı Veri Girişi

Öğretmen Ders Girişi Veri Girişi Öğretmen Grafiği İpucularını göster

Bölüm: Bilgisayar

Kapat

Numara: Öğretmen: Ali Tan, Alparslan Durmaz, Altan Ural, Avni Er, Aziz Bektaş, Doğan Boz, Hakan AK, Salih Zeki Taşçı, Suat Özsarac

Ders Kodu: AAĞS, ADU, AEND, AT16, BGIR, BITKN, DEL2, DEL3, DOM2

Sube: A10C, A11C, A12C, A9C, A9D, M10F

Atölyeler: AgSi, BGir, DiJA, Elek

✓ Ders ve Lab Aynı Yerde Olacak
✓ Lab Bulunacak (A ve B Grubu)
✓ Teorik Bulunacak (1. ve 2. Blok)

Liste Tipi: Sadece Atölye, Tüm Derslikler

Dahil	Baş	Öğretmen	Diğer Öğ	Ders Kodu	Ders Adı	Teori	LAB	Şube1	Şube2	Derslik	Yerleşmiş
True	AE	Avni Er	AU	DÜ9	Dijital Atölyesi	5	8	T10C	-	DiJA	True
True	TB	Tarik Bağrıyanık	SÖ	ADU	Dijital Atölyesi	8	0	M10F	-	DiJA	True
True	AB	Aziz Bektaş	HA	DÜ9	Dijital Atölyesi	5	9	T10D	-	DiJA	True
True	AT	Ali Tan	MIŞ5	Mikroişlemci ve Asm	5	0	M10F	-	Pro2	True	
True	SuÖ	Suna Özkalay	AD	ADU	Dijital Atölyesi	8	0	M10G	-	DiJA	False

23:23:25 01.02.2006

Resim 32b – Veri girişi

Otomatik Ders Programı Hazırlama

Toplam Ders Sayısı = 57+27 saat
Toplam Öğretmen Sayısı = 5+4 kişi

Başla

Unvana Göre

✓ Yerleşmiş verileri sil

İşlem Durumu

100%

Yerleşmeyenlerin Bilgileri

Başlangıç = 23:24:15
5 SuÖ 'in ADU dersi (M10G) atanamadı. (uygun saat bul)
1 saniyede yerleştirme bitti...

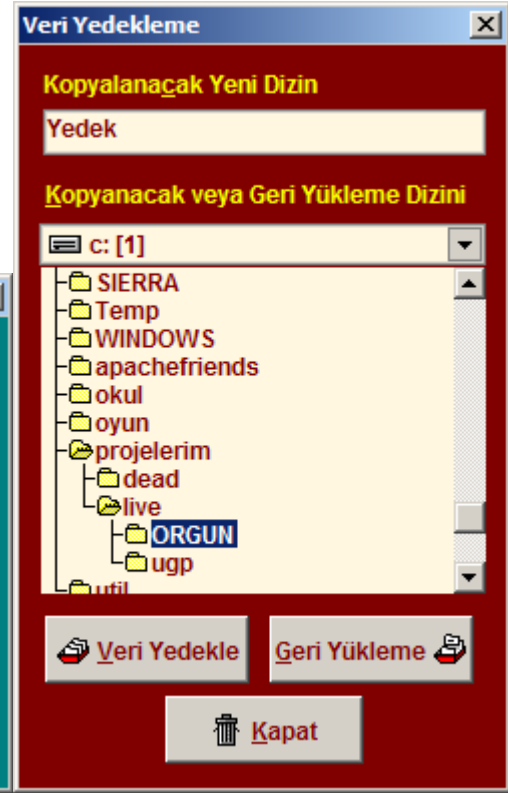
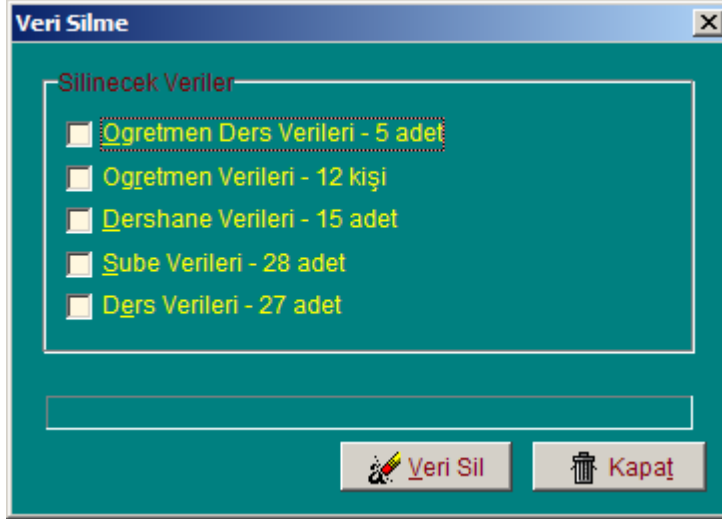
Yerleşenlerin Bilgileri

1 AE 'in DÜ9 dersi
1 AE 'in DÜ9 dersi
1 AE 'in DÜ9 dersi
2 TB 'in ADU dersi
3 AB 'in DÜ9 dersi
3 AB 'in DÜ9 dersi
4 AT 'in MIŞ5 dersi

Tümünü Temizle

Kapat

Resim 32c – Otomatik Program Hazırlama



El ile program düzeltme 2

Öğretmen | Dershane | Şube

BasHarfler	Gun	GunSir	Ders1	Ders2	Ders3	Ders4	Ders5	Ders6	Ders7	Ders8	Ders9	Ders10
AB	Pazartesi	1						DiJ9	DiJ9	DiJ9	DiJ9	DiJ9
AB	Sali	2										
AB	Carsamba	3										
AB	Persembe	4										
AB	Cuma	5	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	
AD	Pazartesi	1										
AD	Sali	2										
AD	Carsamba	3										
AD	Persembe	4										
AD	Cuma	5										
AE	Pazartesi	1	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	
AE	Sali	2	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	
AE	Carsamba	3	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	DiJ9	
AE	Persembe	4										
AE	Cuma	5										
AT	Pazartesi	1	MİŞ5	MİŞ5	MİŞ5	MİŞ5	MİŞ5					
AT	Sali	2										

Excel'e | Arama | Bakım Yap | Verileri Sil | Tazele | Seçenekler | Kapat

Öğretmen: AB-HA | Derslik: DiJA | Şube: T100 | Ders kaydında sorun yok

El ile Ders Yerleştirme Formu

Dershane Kodu: DİJA
Mevcut: 30-Dijital Atelyesi

Ders Ko...	Öğretmen	Sınıf	Yardı...
ADIJ	Suna Özkalay	M10G	AD
ADIJ	Tanık Bağrıyanık	M10F	SÖ
DIJ9	Avni Er	T10C	AU
DIJ9	Aziz Bektaş	T10D	HA
MIŞ5	Ali Tan	M10F	-

PAZARTESİ **SALI** **ÇARŞAMBA** **PERŞEMBE** **CUMA**

	KOD	S1	S2	S3	S4	KOD	S1	S2	S3	S4	KOD	S1	S2	S3	S4	KOD	S1	S2	S3	S4	KOD	S1	S2	S3	S4
DERS1	DIJ9	T10C				DIJ9	T10C				DIJ9	T10C				ADIJ	M10I				DIJ9	T10C			
DERS2	DIJ9	T10C				DIJ9	T10C				DIJ9	T10C				ADIJ	M10I				DIJ9	T10C			
DERS3	DIJ9	T10C				DIJ9	T10C				DIJ9	T10C				ADIJ	M10I				DIJ9	T10C			
DERS4	DIJ9	T10C				DIJ9	T10C				DIJ9	T10C				ADIJ	M10I				DIJ9	T10C			
DERS5	DIJ9	T10C				DIJ9	T10C				DIJ9	T10C				ADIJ	M10I				DIJ9	T10C			
DERS6						DIJ9	T10C				DIJ9	T10C				ADIJ	M10I				DIJ9	T10C			
DERS7						DIJ9	T10C				DIJ9	T10C				ADIJ	M10I				DIJ9	T10C			
DERS8						DIJ9	T10C				DIJ9	T10C				ADIJ	M10I				DIJ9	T10C			
DERS9						DIJ9	T10C				DIJ9	T10C								DIJ9	T10C				
DERS10																									

Rapor Dokum Formu

Öğretim Yılı: 2005-2006

Sube: A10C

Dershane: AgSi

Öğretmen: Ali Tan

Bölüm: Bilgisayar

Rapor Bilgileri **Kapat**

Ekrana **Yazıcıya**

Resim 32d – El ile yerleşim ve raporlama pencereleri

Bu sürümünü lise ders programına göre özelleştirdim. Yani üniversitede günlük 12 ders var iken lisede 10 ders saati var olacak şekilde değişiklik yaptım. Cumartesi de ders yapılabilirdi. Böyle tasarlamamız daha ileriye dönük bir çalışma olmasını sağlamış, ders saati, gün eklemek zor olurdu.

O yıllarda Türkçe karakter, bazı bilgisayarlarda kötü görünüm veriyordu, bazı menülerde Türkçe karakterler var iken bazılarında ı yerine i kullanınca biraz tuhaf olmuş. Elimden geldiğince bu hataları düzelttim. En büyük sorunumuz ise 800x600 ekranda tasarlanan programın yüksek çözünürlükte biraz yana kayması, ona benzer garip pencereler olmasına sebep oldu. Ama pencere ayarlarını aklında tutması sayesinde bu sorunu aşıyoruz. Genelde Teal – koyu yeşil renk ve Arial yazıtipi kullanıldı.

Bunun dışında yedekleme penceresinde “Geri Yükleme”ye güvenmemek lazım. Program o sırada çalıştığı için veritabanları üzerine yeni dosyalar yüklemeyiz. O yüzden en iyisi program kapalı iken, elle yedekleri geri almak.

Program yapmak için 3 yöntemimiz var. Otomatik, el ile 1 ve el ile 2. Otomatik yerleştirme öncesinde belli öğretmenlerin dersi programa katılmayabilir. Bu veri giriş penceresinde “Dahil etme” alanında yapılıyor. İstenirse de el ile 1’de listeden bir ders sağ tıklanıp Otomatik yerleştirilebiliyor. Bu iki pencerede ders programı hazırlama belli kurallar içinde olduğundan, 3. yöntemi kullanabiliyoruz. El ile 2’de veriler tamamen serbestçe değiştirilebilir. Yine de arka planda ders kaydında sorun olup olmadığı, eksiklik var mı diye bir kontrol de var. Bu durum çubuğunda belirtiliyor. Mesela Öğretmen ve Derslik’e ders girilip de şubeye atanmamışsa bu durum çubuğunda rahatlıkla görülüyor. Ders kodlarının tam yazılması gerekli. Küçük büyük harf farkı var. Mesela DİJ9 aynen yazılırsa döküm kusurlu çıktı vermiyor.

Bir derse birden fazla öğretmen de atanabiliyor. İleride gerekirse daha fazla da ekleyebiliriz. Yani 3 veya 4 öğretmen de olabilir. Sadece Dershane dökümünde iki öğretmen görülmüyor.

Programın çalışmasında, açılış ekranından veritabanlarına kadar hiç güvenlik önlemi alınmamış. Yani veriler şifreli değil, kullanıcı hesapları yok. Zaten en büyük eksiklik ağ desteğinin olmaması. Program sadece bir bilgisayarda çalışsın diye düşünüldü. Excel'e aktarım ile birazcık (tek yönde) dış dünyaya da uyumlu hale geliyor.

Sıkıştırılınca kaynak kod ile birlikte 870KB oluyor. Şu adresten ibinstall.defined.net indirecek BDE kurulumu ile hemen her Windows sürümünde programımız çalışıyor. Kurulmadan önce toplam gereken alan 10MB ediyor. BDE kurulunca el ile "orgun" aliası eklenir ve program çalışmaya hazır hale gelir. Alias yapılırken programın bulunduğu yerdeki "I__C_DONEM" klasörü belirtilir. Aslında Installshield ile kurulum hazırlasaydım, bu dert olmayacaktı. Neyse, bedava işin desteği az oluyor işte.

"Kritik hata" mesajları sayesinde büyük hatalarda, programın kapanıp gitmesi engelleniyor. Bazı hallerde veri diske kaydedilemeden, program kapanabilir. Ve tüm yapılan değişiklikler unutulabilir. O sebeple program belli aralıklarla kapatılıp tekrar açılmalıdır.

Minimum gereklilikler: Win98, 32MB RAM, 10MB disk alanı, 4MB ekran kartı, fare, yazıcı
Tavsiye Edilen: WindowsXP, 512MB RAM, 64MB ekran kartı, lazer yazıcı

Özellikler:

- 1- Ders verilerini girme, ders atama, istenmeyen gün ve saat seçme
- 2- Ders bölme, şube birleştirme, atölye grubu oluşturma,
- 3- a- Otomatik ders hazırlama (farklı önceliklere göre, şube ve derslik kapasitesine göre bakılarak)
b- El ile düzenleme (sürükle bırak yöntemi, tek dersi otomatik yerleştirme)
c- Tamamen kısıtlamasız el ile müdahale ve düzeltme formu
ç- Çakışma ve yerleşmeyen bilgileri listeleme, renklendirme
- 4- Yazıcı dökümü, tüm sınıfların dökümü, Excel'e veri çıkışı
- 5- Teneffüs saatleri düzenlenebilir, okul adı, öğrenim yılı değiştirilebilir
- 6- Veri yedekleme, silme, arama, sıralama, bakım, grafik gösterim
- 7- Windows standart görünümünde rahat kullanım özellikleri (ipuçları, kısayollar, pencere boyutları saklama, basitlik, okunaklılık)
- 8- Kod olarak Cumartesi ve 12 ders saati desteği hazırdır. İstenirse tek programdan birçok okul ve bölüm de yapılabilir.

Uyarılar:

- * Yazıcı çıkışının düzgün olması için bir yazıcı tanımlı olmalıdır.
- ** Excel çıkışı Microsoft Excel kurulu ise çalışır.
- *** Bilgisayarınızda o anda çalışan diğer programlar, performansı etkiler.

Okul Ders Programının Algoritması: (bu kısmı hiçbir değişiklik yapmadan aktardım)

Tüm programlama dillerinde bir uygulama geliştirmeye başlamadan önce yapılacak program için bir algoritma hazırlamak gereklidir. Oluşturulacak bu algoritma program yazılmaya başlandığında büyük fayda sağlar. Programcı programın neler yapması gerektiğini, programın çalışma mantığını, ilerleme sırasını algoritmayı hazırlarken belirleyeceğinden programın yazılış aşaması daha kolay ve hızlı bir şekilde gerçekleşir. Bu nedenle bu bölümde Okul Ders Programının algoritması açıklanmıştır.

Okul Ders Programı algoritması dört ana bölüm halinde düşünülmüştür. Bu bölümler veri girişi, otomatik ders programı hazırlama, el ile ders programı hazırlama ve raporlamadır. Diğer bölümlerin dayandığı temel bölüm veri girişi olduğundan önce bu bölümün algoritması hazırlanmıştır.

Veri Girişi Algoritması

Yukarıda belirtildiği gibi veri girişi bölümü temel bölümdür. Bu bölümde diğer bölümlerin kullanacağı ve programın görevini yerine getirebilmesi için gerekli bilgilerin bilgisayara girilmesi gerçekleştirilir. Veri girişi bölümü de kendi içinde iki kısma ayrılır. Program okul ders programını hazırlayacağına göre ilk kısımda öğretim elemanları bilgileri, şube bilgileri, dersane bilgileri ve ders bilgileri girilir. Bu kısma genel veri girişi de denilebilir. İkinci kısımda

ise ilk kısımda girilen bilgiler kullanılarak öğretim elemanlarının okutacakları dersler, bu dersleri alacak şubeler ve dersin programa yerleşmesinde istenecek bazı özel koşullar belirlenir. Bu kısım da öğretim elemanları ders girişi olarak adlandırılabilir. Aşağıda bu kısımlar daha ayrıntılı olarak anlatılmıştır.

Genel Veri Girişi

Yukarıda belirtildiği gibi bu kısımda temel veriler olan öğretim elemanları, şube, dersane ve ders bilgileri girilir. Bu veriler oluşturulan veritabanı tablolarında tutulur.

Öğretim Elemanları Verileri

Bu bölümde öğretim elemanlarının ad ve soyadları, unvanları, isim ve soyadlarının baş harfleri ve her öğretim elemanı için bir numara girilir. Ayrıca her öğretim elemanının tercihinine göre boş olmasını istediği saatleri işaretleyebilmesi ve o saatlere ders programında ders konulmaması sağlanabilir. Bu ders saati kilitleme işlemi ders günleri olan Pazartesi, Salı, Çarşamba, Perşembe, Cuma ve Cumartesi günleri için 12 ders saati üzerinden düşünülmüştür.

Ad Soyad	Unvan	No	Baş Harfler	Pazartesi 1	Pazartesi 2	...	Pazartesi 12	Salı 1	...	Cumartesi 12
Ali Aksoy	1	1	AA	True	True	...	False	False	...	False
Ahmet Ok	4	2	AO	False	False	...	False	False	...	False

Tablo 1 - Öğretim elemanları verileri tablo yapısı

Yukarıdaki veritabanı yapısına istenilen sayıda kayıt girilebilir. Ad Soyad alanına öğretim görevlisinin tam adı girilir. Unvan alanında Prof.Dr. 1 ile, Doc.Dr. 2 ile, Yrd.Doc.Dr. 3 ile, Öğretim Görevlisi 4 ile ve Araştırma Görevlisi 5 ile belirtilir. Ders programı hazırlanırken öncelik yüksek unvanlı öğretim elemanlarının derslerine tanınacağı için unvan alanı önemlidir. Kolaylık sağlaması açısından her öğretim elemanına bir numara verilir. Baş Harfler alanında öğretim elemanının ad ve soyadının baş harfleri girilir. Aynı baş harflere sahip birden fazla kayıt varsa soyadın ikinci harfi de baş harflerin sonuna eklenebilir. Pazartesi1'den başlayıp Cumartesi12'ye kadar giden alanlar ise önceden belirtildiği gibi ders saati kilitleme olanağı sağlar. Bu alanlar Pazartesi, Salı, Çarşamba, Perşembe, Cuma ve Cumartesi günlerinin her biri için 1' den 12' ye olmak üzere 12' şer adettir. (Pazartesi1, Pazartesi2, ..., Pazartesi12, Salı1, Salı2, ..., Salı12, ...) Bu alanlar mantıksal değerler alır ve varsayılan değerleri Yanlış yani False' dır. Bunun anlamı o saate ders yerleştirilebilir demektir. Eğer alanın değeri Doğru yani True ile değiştirilir ise o saate ders yerleştirilmesi önlenir. Örneğin Tablo 1'de görüldüğü gibi Ali Aksoy isimli öğretim görevlisi için Pazartesi1 ve Pazartesi2 alanlarına True değeri verilerek ders programı hazırlanırken adı geçen kişiye Pazartesi günü ilk iki saate ders konulmaması sağlanabilir. Eğer kişi tüm bir günü kilitlemek isterse o güne ait 12 alana da True değeri verilerek bu sağlanabilir.

Bu tablo ile öğretim elemanlarının bilgileri kayıtlar halinde tutulabilir. Ancak ders programı hazırlanırken her bir öğretim elemanının haftalık ders programlarını ayrı ayrı tutacak başka bir tablo yapısına daha ihtiyaç vardır. Yani her bir öğretim elemanının Pazartesi günü hangi saatte hangi dersi bulunuyor, yine Salı günü hangi saatte hangi dersi var gibi bilgileri tüm ders günleri için tutan bir tablo daha oluşturmak gereklidir.

Baş Harfler	Gün	Ders 1	Ders 2	Ders 3	Ders 4	Ders 5	Ders 6	Ders 7	Ders 8	Ders 9	Ders 10	Ders 11	Ders 12
AA	Pazartesi												
AA	Salı												
AA	Çarşamba												
AA	Perşembe												
AA	Cuma												
AA	Cumartesi												
AO	Pazartesi												
AO	Salı												

Tablo 2 - Öğretim elemanı ders programı tablosu

Bu ikinci tablo ilk tablo ile koordine olarak çalışmalıdır. Yani ilk tabloya eklenen her bir öğretim elemanı için ikinci tabloda otomatik altı ayrı kayıt açılır. Bu altı kayıt ilk tabloya girilen öğretim elemanının baş harflerine dayanır ve her kayıt bir ders günü için açılır. Yine Tablo 2'de görüldüğü gibi her bir kayıt on iki ders saati içerir. Bu iki tablo yapısı arasındaki ilişki Tablo 3'de daha iyi görülebilir.

Ad Soyad	unvan	No	Baş Harfler	Pazartesi 1	Pazartesi 2	..	Pazartesi 12	Salı 1	..	Cumartesi 12
Ali Aksoy	1	1	AA	True	True	..	False	False	..	False
Ahmet Ok	4	2	AO	False	False	..	False	False	..	False

Baş Harfler	Gün	Ders 1	Ders 2	Ders 3	Ders 4	Ders 5	Ders 6	Ders 7	Ders 8	Ders 9	Ders 10	Ders 11	Ders 12
AA	Pazartesi												
AA	Salı												
AA	Çarşamba												
AA	Perşembe												
AA	Cuma												
AA	Cumartesi												
AO	Pazartesi												

Tablo 3 - Öğretim elemanları tabloları arasındaki bağlantı

Tablo 3'deki mantığa göre ilk tabloda herhangi bir öğretim elemanının kaydı silindiğinde, o kişinin ikinci tablodaki altı kaydı da silinir. Böylece çıkabilecek karışıklıklar engellenir. Yine aynı şekilde birinci tabloda bir kayıta yapılacak baş harf değişikliği ikinci tablodaki ilgili kayıtlara da otomatik olarak uygulanır. İkinci veritabanı tablosundaki Ders1'den Ders12'ye kadar olan alanlara ders programı hazırlanırken ilgili öğretim elemanının dersleri yerleştirilir. Bu aşama ileride açıklanacaktır.

Şube Verileri

Şube verileri bölümü öğretim elemanı verileri kısmı ile hemen hemen aynıdır. Veriler yine iki ayrı veritabanında tutulur. İlk tabloda şube adı, şubenin mevcudu ve bölüm bilgileri kayıtlıdır. Yine öğretim elemanları verilerinde olduğu gibi şubeler için de ders saatleri kilitleme imkanı sağlanmıştır. Yani tüm ders günleri için 12'şer mantıksal alan ayrılmıştır. Bu alanlar Pazartesi1'den başlayıp Cumartesi12'ye kadar gider.

Şube Adı	Mevcut	Bölüm	Pazartesi 1	Pazartesi 2	...	Pazartesi 12	Salı 1	...	Cumartesi 12
1BBK	45	Bilgisayar	True	True	...	True	True	...	True
2B1	43	Bilgisayar	False	False	...	False	False	...	False

Tablo 4 - Şube verileri yapısı

Şube adı alanında şubelerin isimleri tutulur. Mevcut alanı ders programı hazırlanırken uygun dersane bulmakta kullanılır. Bölüm alanında ise şubenin dahil olduğu bölüm belirtilir. Bölüm bilgisi ders programının raporlanması aşaması için gereklidir. Öğretim elemanı verilerinde açıklandığı gibi Pazartesi1'den Cumartesi12'ye kadar olan alanlara True değerleri verilerek ders programında ilgili şubenin o saatine ders konulmaması sağlanabilir.

Aynı şekilde her bir şubenin haftalık ders programının tutulacağı ikinci bir tablo yapısı da gereklidir. Bu yapı da öğretim elemanları verilerindeki ile aynıdır.

Şube	Gün	Ders 1	Ders 2	Ders 3	Ders 4	Ders 5	Ders 6	Ders 7	Ders 8	Ders 9	Ders 10	Ders 11	Ders 12
1BBK	Pazartesi												
1BBK	Salı												
1BBK	Çarşamba												
1BBK	Perşembe												
1BBK	Cuma												
1BBK	Cumartesi												
2B1	Pazartesi												
2B1	Salı												

Tablo 5 - Şubeler için ders programı tablosu

İlk tabloya girilen her yeni şube için ikinci veritabanında otomatik olarak altı kayıt açılır. Bu altı kayıttan her biri bir ders günü içindir. İki veritabanı birbirine bağlıdır. Yani ilk tablodan silinen bir şubenin kayıtları ikinci tablodan da silinir. Aynı şekilde ilk tabloda şube adında yapılan değişiklikler aynen ikinci veritabanına da yansır. Ders1'den Ders12'ye olan alanlara ders programı hazırlanırken şubenin dersleri doldurulur.

Dershane Verileri

Dershane verilerinin algoritması da önceki iki bölüm ile aynıdır. Birinci veritabanı tablosunda dershane kodu, dershane mevcudu ve açıklama bilgileri depolanır. Dershane kodu dershane adı olarak düşünülebilir. Mevcut ders programı oluşturulurken şube mevcutları ile karşılaştırmada kullanılır. Açıklama ise isteğe bağlıdır. Dershane hakkında açıklayıcı bilgi vermekte kullanılır. Örneğin anfi veya laboratuvarlar belirtilir. Ders saati kilitleme yine aynıdır.

Dershane Kodu	Mevcut	Açıklama	Pazartesi 1	Pazartesi 2	...	Pazartesi 12	Salı 1	...	Cumartesi 12
D410	45	Sınıf	True	True	...	True	True	...	True
D510	40	Sınıf	False	False	...	False	False	...	False

Tablo 6 - Dershane verileri yapısı

İkinci veritabanı ile de tüm dershanelerin ayrı ayrı haftalık programları tutulur. Diğer iki kısımda olduğu gibi yine ilk tabloya girilen yeni kayıt için ikincide otomatik olarak altı gün için ayrı kayıtlar açılır. Yine ilk tablodaki değişiklikler ikinciye otomatik yansır.

Dershane	Gün	Ders 1	Ders 2	Ders 3	Ders 4	Ders 5	Ders 6	Ders 7	Ders 8	Ders 9	Ders 10	Ders 11	Ders 12
D410	Pazartesi												
D410	Salı												
D410	Çarşamba												
D410	Perşembe												
D410	Cuma												
D410	Cumartesi												
D510	Pazartesi												
D510	Salı												

Tablo 7 - Dershaneler için ders programı tablosu

Ders Verileri

Ders verilerinin algoritması diğerlerinden daha basittir. Tek bir veritabanı tablosu ile ders adı, ders kodu, teori ve laboratuvar saatleri sayısı, dersin hangi bölümün dersi olduğu bilgileri saklanır. Ek olarak kaydı yapılan dersin Bitirme Projesi veya Öğretmenlik Uygulaması gibi özel yapıda bir ders olup olmadığını belirtmek için bir alan tanımlanmalıdır. Bu yapı aşağıdaki şekilde oluşturulur.

Ders Kodu	Ders Adı	Teori	Laboratuvar	Bölüm	Özel Ders
EDC482	Öğretmenlik Uyg.	0	8	Bilgisayar	True
ELC114	Comp. Prog.1	3	2	Bilgisayar	False

Tablo 8 - Ders verileri yapısı

Ders kodu alanında derse verilen kod isim belirtilir. Ders adı ile dersin uzun adı verilir. Teori ve laboratuvar alanları dersin teori ve laboratuvar saat sayılarını belirtir. Bölüm dersin hangi bölümün dersi olduğunu belirler. Özel ders alanı ise mantıksal bir alandır. Varsayılan değeri False'dur. Eğer bu alana True değeri verilirse o ders Öğretmenlik Uygulaması gibi sadece şubelerin programına yerleşecek bir ders olarak tanımlanır ve ders programı oluşturulurken buna göre işleme konulur.

Öğretim Elemanları Ders Verileri Girişi

Öğretim elemanları ders verileri kısmında ise ilk bölümde girilen bilgiler kullanılarak öğretim elemanlarının okutacakları dersler, bu dersleri alacak şubeler ve dersin programa yerleşmesinde istenecek bazı özel koşullar belirlenir. Ders programı oluşturulurken kullanılacak asıl bilgiler bu kısımda girilir. Bir önceki bölümde girilen bilgiler tamamen bu kısmın oluşturulmasına yardımcı verilerdir. Öğretim elemanları ders verileri oluşturulan bir veritabanı tablosunda kayıtlar halinde tutulur.

Öğretim Elemanı	Ders Kodu	Ders Adı	Teori	Lab	Şube 1	Şube 2	Şube 3	Şube 4	Ders-Lab. Birleştir	Teori Böl	Lab. Böl	Lab 1	Lab 2
Ahmet Ok	SCI102	Matematik	6	0	2B1				False	True	False		
Ahmet Ok	SCI102	Matematik	6	0	2B2				False	True	False		
Ali Aksoy	ELC114	Comp. Prog.1	3	2	2B1	2B2			True	False	True	D408	D409
Ali Aksoy	ELC332	Comp. Prog. 3	3	2	3BB	3BK	3BT	3BE	True	False	False	D408	D409

Tablo 9 - Öğretim elemanları ders verileri yapısı

Tablo 9'da görülen yapıya her öğretim elemanının okutacağı dersler ayrı kayıtlar olarak girilir. Bu kayıtlara bağlı olarak ders programının nasıl hazırlanacağı Otomatik Ders Programı Algoritması kısmında açıklanacaktır. Öğretim elemanı alanında öğretim görevlisinin ad ve soyadı tutulur. Öğretim görevlisi daha önce anlatılan Öğretim Elemanları Verileri tablosundaki kayıtlar arasından seçilir. Ders kodu öğretim elemanının okutacağı derse verilen kod bilgisini

kapsar. Bu kod bilgisi de önceden kayıtları girilen Ders Verileri veritabanı'ından seçilir. Ders kodu seçildiğinde ders adı, teori ve laboratuvar bilgileri otomatik olarak Ders Verileri tablosundan okunarak Tablo 9'daki ilgili alanlara yazılır. Bir dersin teori kısmı aynı saatte ve dershanede, aynı öğretim elemanı tarafından dört şubeye birden verilebileceğinden Şube1, Şube2, Şube3 ve Şube4 adlı dört alan bulunur. Öğretim görevlisinden dersi aynı saatte ortak alan şubeler sıra ile bu alanlara yerleştirilir. Eğer aynı dersi alan şubeler, bu dersi farklı saatlerde görecekler ise her şube için ayrı kayıt girilmelidir. Örneğin yukarıdaki tabloda görüldüğü gibi Ahmet Ok isimli öğretim görevlisi SCI102 kodu Matematik dersini 2B1 ve 2B2 şubelerine ayrı zamanlarda göstereceğinden her bir şube için aynı öğretim elemanı adına iki ayrı kayıt açılmıştır. Aynı şekilde Ali Aksoy isimli öğretim görevlisi ELC114 kodlu Programlama dersinin 3 saat olan teori bölümünü 2B1 ve 2B2 şubelerine aynı saatte ve aynı derslikte beraber göstereceğinden bu iki şube için sadece bir kayıt açılmıştır. Ancak bu işlem sadece teori saatleri için geçerlidir. Ders programı hazırlanırken laboratuvar saatleri her şube için, teoriler ortak olsa bile, ayrı ayrı yerleştirilir.

Sonra gelen Ders-Lab Birleştir, Teori Böl, Laboratuvar Böl alanları mantıksal alanlardır ve True/False değerleri alırlar. Bu alanların varsayılan değerleri False'dur. Ders-Lab Birleştir alanına True değeri verilmesi dersin teori saatlerinin de laboratuvar saatleri gibi laboratuvar da işleneceğini belirtir. Yine tablodaki örnekte Ali Aksoy'un Programlama 1 ve Programlama 3 dersleri için Ders-Lab Birleştir alanına True değeri verilerek ders programı hazırlanırken teori saatlerinin de laboratuvara yerleştirilmesi sağlanır. Teori Böl alanına True atanması ile ise dersin teori saatlerinin ikiye bölünerek ayrı günlere yerleştirilmesi sağlanır. Örnekte Ahmet Ok adlı öğretim görevlisinin Matematik dersi kayıtlarında Teori Böl alanına True atanarak dersin 6 saat olan teori kısmının üçer saat halinde ayrı günlere yerleştirilmesi sağlanabilir. Son olarak Lab Bölme alanını True yaparak dersin laboratuvar saatlerini A ve B grubu olarak iki gruba ayırmak mümkündür. Tablo 9'da bunu Ali Aksoy'un Programlama 1 dersi kaydında görebilirsiniz.

Son iki alan olan Lab1 ve Lab2 alanlarına ise dersin varsa laboratuvar kısmının gösterileceği laboratuvarların kodu girilerek laboratuvar saatlerinin bu dersliklere yerleştirilmesi sağlanır. Bazı dersler için birden fazla laboratuvar mevcut ise bu alanlara iki ayrı laboratuvar girilebilir. Program hazırlanırken dersin laboratuvar kısmı önce Lab1 alanındaki dersliğe yerleştirilmeye çalışılır. Eğer burası uygun değil ise laboratuvar saatleri Lab2 alanında belirtilen dershaneye yerleştirilir.

Otomatik Ders Programı Algoritması

Otomatik ders programı ile temel veriler (dersler, şubeler, dershaneler, öğretim elemanları veri tabanları) ve öğretim elemanları için girilen ders verileri kullanılarak otomatik ders hazırlanır. Otomatik Program başlayınca sıra ile şunlar olur. Önce derslerin yerleştirildiği veritabanlarının boşaltılıp boşaltılmayacağı sorulur. Bu işlem ile daha önce hazırlanmış bir ders programı var ise silinmesi sağlanır. Özel dersler (öğretmenlik uygulanması, bitirme projesi...) gibi dersler ayıklanıp bir diziye atanır. Özel dersler şöyle yerleştirilir; kaç şube için bu ders belirtilmiş ise her şubeye farklı günlere ders atanır. Yani öğretmenlik uygulaması 4BB için pazartesi, 4BE için salı günü olması sağlanır. İlk kayıt okunup öğretim elemanı başharfleri, ders kodu, şubeler ve diğer bilgiler ara değerlere atanır.

Eğer öğretim elemanının ders kaydında hatalar var ise ekrana hata mesajı gelir. Bu hatalar bilinmeyen şube girmek, bilinmeyen derslik kodu girmek, laboratuvarı olan dersin laboratuvar kodunun verilmemesi, ders kodunun olmaması gibi hatalardır.

Eğer ders özel bir ders değil ise derslik bulma, teori bölme ve uygun saat arama işlemleri yapılır. Özel ders ise derslik ve öğretim elemanları gözardı edildiği için daha basit bir saat kontrolü yapılır.

Teorik bölünecek ise döngü içinde teorik dersin yarısı bir güne, diğer yarısı ayrı bir güne konulur. Eğer bu dersin laboratuvar sayısı sıfır ise yani uygulaması olmayan bir ders ise, veri giriş kısmında laboratuvar1 ve laboratuvar2 olarak girilen kodlar bu dersin teorisinin işleneceği derslik olarak yerleştirilir. Eğer hiç uygulaması olmayan derse laboratuvar da belirtilmemiş ise bu derse derslik aranır. Eğer uygun yer bulunamaz ise ekrana hata mesajı eklenir. Eğer dersin uygulaması var ise, ders ve laboratuvarın aynı derslikte olacağı işaretli mi kontrol edilir. İşaretli değil ise ders için derslik aranır. Bu ders için belirtilmiş her şube için ayrı ayrı laboratuvarlar yerleştirilir, fakat hepsi aynı derslikte aynı saatte teorik derslerini işlerler. Ayrıca laboratuvar bölme (A ve B grubu gibi) kutusu işaretlenmiş ise her şubenin ayrı ayrı laboratuvar grupları farklı saatlerde olmak üzere uygulamalarını yaparlar. Bilgisayar eğitimi dersi 3BB ve 3BK şubelerine verilecek ve bu ders için laboratuvar bölme var ise 3BB'nin uygulaması ikiye, 3BK'nın uygulaması farklı saatlerde yine uygulaması ikiye ayrılır. Fakat her iki şube de aynı derslikte teorilerini işlerler.

Uygun saat arama ve derslik bulma kullanılan iki ana prosedürdür. Her çağrıldıklarında haftanın ilk günü ilk saatinden itibaren işlemler ilerler. Örgün eğitimde, dersler yerleştirilirken uygulamalar için sınır 12 saat iken, teorik için bu 8 saattir. Yani 8. saatten sonra sadece laboratuvar konulur. İkinci öğretim için her iki sınır 8'dir. Buradaki amaç gündüz öğrencilerin son saatlerde derslikleri kullanmayıp ikinci öğretim derslik imkanını rahatlatmaktır.

Öğretim elemanına girilen dersler yerleştirilirken öğretim elemanın ünvanına göre sıralanır. Bölüm ayrımı yapmaksızın dersleri yerleştirir. Yani bilgisayar bölümü ile elektrik bölümünün dersleri aynı anda yerleştirilir. Bunun anlamı bölüm ayrımının sadece ders kodu ve şube için geçerli olduğudur. Böylece farklı bölüm öğrencileri aynı binada aynı derslikleri paylaşırlar. Asla derslikleri, öğretim elemanları ve şubeleri çakışmaz.

Veri girişinde ayrıca belli saatlerin işaretlenip bu saatlere ders konulmaması istenebildiği için o saatlere ders yerleştirilmez. Örneğin D410 dershanesi pazartesi gününün tümü işaretlenmiş ise bu saatlerde ders burada işlenmez. Bu sayede derslik başka amaçlar için kullanılabilir. Bu öğretim elemanları ve şubeler için de geçerlidir. İstenen şubeye istenen saatlerde, istenen öğretim elemanına istenen saatlere ders konulmaz.

Öğretim elemanı, ilk şube, ikinci şube, üçüncü şube, dördüncü şube, ilk derslik ve ikinci derslik aynı gün ve saatte kilitli değilse ve dolu değil ise yerleşme işlemi yapılır. Birisinde bile uygun olmayan durum olduğu anda gün ve saat değerleri değiştirilmeye devam edilir. Gün ve saat bittiği halde hala ders yerleşmemiş ise, ana prosedür olan başlangıca dönülüp, ekrana hata tipi ve açıklaması eklenir.

Boş dersane ararken kullanılan mevcut şu şart kullanılarak denetlenir; eğer şubelerin toplam mevcutları dershanenin kapasitesinin 30 eksiğinden büyük veya eşit, dershanenin kapasitesinin 20 fazlasından küçük veya eşit ise bu derslik uygun demektir. Böylece örneğin 24 mevcutlu 4BB'nin dersi, 200 kişi kapasiteli D102 dersliğine konulmaz.

El ile Ders Programı Hazırlama Algoritması

Bu bölümde de Veri Girişi bölümünde girilen bilgiler kullanılır. El ile Ders Programı Hazırlama bölümünün algoritması dershanelerin haftalık ders programlarına dayanır. Yani ekrana seçilen bir dershanenin haftalık ders programı çizelgesi gelir. Her gün için on iki ders saati vardır ve her ders saatinin yanında dört şube için ayrılmış alanlar vardır. Bunun nedeni daha önce belirtildiği gibi dört ayrı şubenin aynı dersi aynı anda alabilme olasılığıdır. Dershaneler için haftalık program çizelgesi Tablo 10'da ki şekildedir.

D410	PAZARTESİ					SALI					...
	Ders Kodu	Şube1	Şube2	Şube3	Şube4	Ders Kodu	Şube1	Şube2	Şube3	Şube4	
Ders 1	SCI102	2B1									...
Ders 2	SCI102	2B1				ELC332	3BB	3BK	3BT	3BE	...
Ders 3	SCI102	2B1				ELC332	3BB	3BK	3BT	3BE	...
Ders 4	SCI102	2B1				ELC332	3BB	3BK	3BT	3BE	...
Ders 5	SCI102	2B1									...
Ders 6	SCI102	2B1									...
Ders 7											...
Ders 8											...
Ders 9											...
Ders 10											...
Ders 11											...
Ders 12											...

Tablo 10 - Dersane program çizelgesi

Tablo 10'daki çizelgede sadece Pazartesi ve Salı günleri görülmektedir. Ancak doğal olarak çizelge tüm ders günlerini kapsar. Yukarıda örnek olarak bu iki gün kullanılmıştır.

Çizelgede Ders Kodu alanlarından birisine bir dersin kodu girildiğinde o dersi alan şubelerin bir listesi ile karşılaşılır. Bu listeyi oluşturmak için Veri Girişi bölümünde girilen Öğretim Elemanları Ders Verileri kayıtları kullanılır. Girilen ders kodu ile kayıtlardaki ders kodları karşılaştırılarak uygun şubeler dersi ortak alma durumları da göz önüne alınarak listelenirler. Böylece Şube alanlarını doldurma işlemi bu listeden seçilerek yapılabilir. Bu şekilde el ile girilen bilgiler ders programlarına kaydedilir.

Veri Girişi kısmında açıklandığı gibi her yeni öğretim elemanı kaydı için öğretim görevlilerinin haftalık ders programını tutacak olan veritabanı tablosunda altı kayıt otomatik olarak açılır (Tablo 2'deki yapı). Yine hatırlanacağı gibi her bir kayıt bir ders günü içindir. Aynı işlem şubeler ve dershaneler için de geçerlidir. İşte dershanelerin ders programı çizelgesine el ile girilen bilgiler sözü edilen veri tabanlarında açılan bu kayıtlarda ilgili ders saatleri alanlarına yerleştirilir. Bu kayıt işlemi Tablo 10'da ki SCI102 kodlu ders için aşağıda açıklanmıştır.

Tablo 10'da D410 dershanesine Pazartesi günü ilk altı saat için SCI102 kodlu Matematik dersi yerleştirilmiştir. Bu dersi 2B1 şubesi tek olarak alacaktır. Bu bilgiler veri tabanlarına kaydedilirken öncelikle dershanelerin haftalık ders programını tutan veritabanı'nda D410 dershanesinin Pazartesi günü için açılan kaydı bulunur. Kayıttaki Ders1, Ders2, Ders3, Ders4, Ders5 ve Ders6 adlı alanlara bakılır. Eğer bu alanlardan bir yada fazlası başka bir ders kodu ile dolu ise, yani çakışma var ise kayıt yapılmaz ve uyarı mesajı verilir. Ardından şubelerin ders programını tutan veritabanı

tablosunda 2B1 şubesinin Pazartesi günü için açılmış kayıt bulunur ve Ders1'den Ders6'ya olan alanlar için çakışma olup olmadığı kontrol edilir (Eğer birden fazla şube dersi ortak olarak alırsa her bir şubenin ilgili kayıtları sıra ile kontrol edilir). Son olarak öğretim elemanları ders verilerini tutan veritabanı'ndan SCI201 kodlu Matematik dersini 2B1 şubesine göstermekle görevli öğretim elemanının adı ve baş harfleri öğrenilir. Örnekte görevlinin adının Ahmet Ok ve baş harflerinin AO olduğu varsayılmıştır. Aynı şekilde öğretim görevlilerinin ders programını tutan veritabanı tablosunda ilgili kişinin Pazartesi günü için açılmış kaydı bulunur. Ders1'den Ders6'ya olan alanlar kontrol edilir. Tüm bu kontrollerin sonucunda eğer herhangi bir çakışma yoksa dersin kodu olan SCI102 bu üç veritabanı tablosunda ilgili kayıtların Ders1'den Ders6'ya kadar olan alanlarına kaydedilir. Eğer çakışmalar bulunur ise bunların bir listesi kullanıcıya bildirilir ve kayıt yapılmaz. Daha sonra ders programı açık olarak görülmek istendiğinde bilgiler bu üç dosyadan okunarak birleştirilir. Dersin veri tabanlarına kaydedilmiş hali aşağıdaki şekildedir.

Dershane	Gün	Ders 1	Ders 2	Ders 3	Ders 4	Ders 5	Ders 6	Ders 7	Ders 8	Ders 9	Ders 10	Ders 11	Ders 12
D410	Pazartesi	SCI102	SCI102	SCI102	SCI102	SCI102	SCI102						
D410	Salı												
D410	Cumartesi												

a- Dershane tablosuna kayıt

Şube	Gün	Ders 1	Ders 2	Ders 3	Ders 4	Ders 5	Ders 6	Ders 7	Ders 8	Ders 9	Ders 10	Ders 11	Ders 12
2B1	Pazartesi	SCI102	SCI102	SCI102	SCI102	SCI102	SCI102						
2B1	Salı												
2B1	Cumartesi												

b- Şube tablosuna kayıt

Öğretim Elemanı	Gün	Ders 1	Ders 2	Ders 3	Ders 4	Ders 5	Ders 6	Ders 7	Ders 8	Ders 9	Ders 10	Ders 11	Ders 12
AO	Pazartesi	SCI102	SCI102	SCI102	SCI102	SCI102	SCI102						
AO	Salı												
AO	Cumartesi												

c- Öğretim elemanı tablosuna kayıt

Tablo 11 - Dersin veri tabanlarına kaydı

Tablolarda da görüldüğü gibi SCI201 dersi öğretim elemanının, Dershanenin ve şubenin Pazartesi günü ilk altı saatine kaydedilmiştir. Bu üç tablodan veriler okunup karşılaştırılarak birleştirilerek öğretim elemanı, şube ve dershane verilerinin tamamının olduğu tam bir ders programı hazırlanabilir.

Raporlama Bölümü Algoritması

Raporlama bölümünde otomatik veya el ile hazırlanan ders programlarının dökümleri alınır. Raporlama dört tipte olabilir. Şube ders programı, öğretim elemanı ders programı, dershane ders programı ve genel ders programı. Bu dökümleri oluşturmak için öğretim elemanlarının ders programlarını tutan veritabanındaki bilgiler, şubelerin ders programlarını tutan veritabanındaki bilgiler ve dersliklerin haftalık programlarını tutan veritabanındaki bilgiler karşılaştırılır ve birbirlerini tutanlar birleştirilir.

Şube için ders programı raporunda ders, öğretim elemanı ve derslik bilgileri vardır. Bu döküm oluşturulurken şubelerin ders programlarını tutan veritabanından istenilen şubenin Pazartesi, Salı, Çarşamba, Perşembe, Cuma ve Cumartesi günleri için açılmış kayıtları esas alınır. Öncelikle şubenin Pazartesi gününe ait kaydın Ders1 alanındaki ders kodu ele alınır. Bu ders kodu öğretim elemanlarının programlarını tutan veritabanı dosyasındaki Pazartesi gününe ait tüm kayıtlar ile karşılaştırılır. Pazartesi gününe ait kaydının Ders1 alanındaki ders kodu bilgisi şubeninki ile aynı olan öğretim görevlisi şubenin raporuna yazılır. Aynı işlem derslik için de tekrar edilir. Pazartesi günü kaydının Ders1 alanındaki bilgi aranılan ders kodu ile birbirini tutan dershane dersin işleneceği dershane demektir. Pazartesi günündeki tüm ders saatlerinin ders kodları, öğretim görevlileri ve dershaneleri bulununca diğer günlere geçilir.

Dershane raporunun mantığı da aynı şekildedir. Dershane programı dökümünde ders kodu, dersi alan şubeler ve öğretim görevlisi bilgileri bulunur. Bu raporda istenilen dershane programı esas alınır ve yukarıda açıklanan yöntem ile her ders saati için şubeler ve öğretim elemanı bilgileri bulunarak raporda yazdırılır. Tek fark dersler dört şubeye kadar ortak görülebildiğinden şubelerin veritabanı dosyası dört kere taranır ve aranan ders koduna uygun tüm şubeler bulunur.

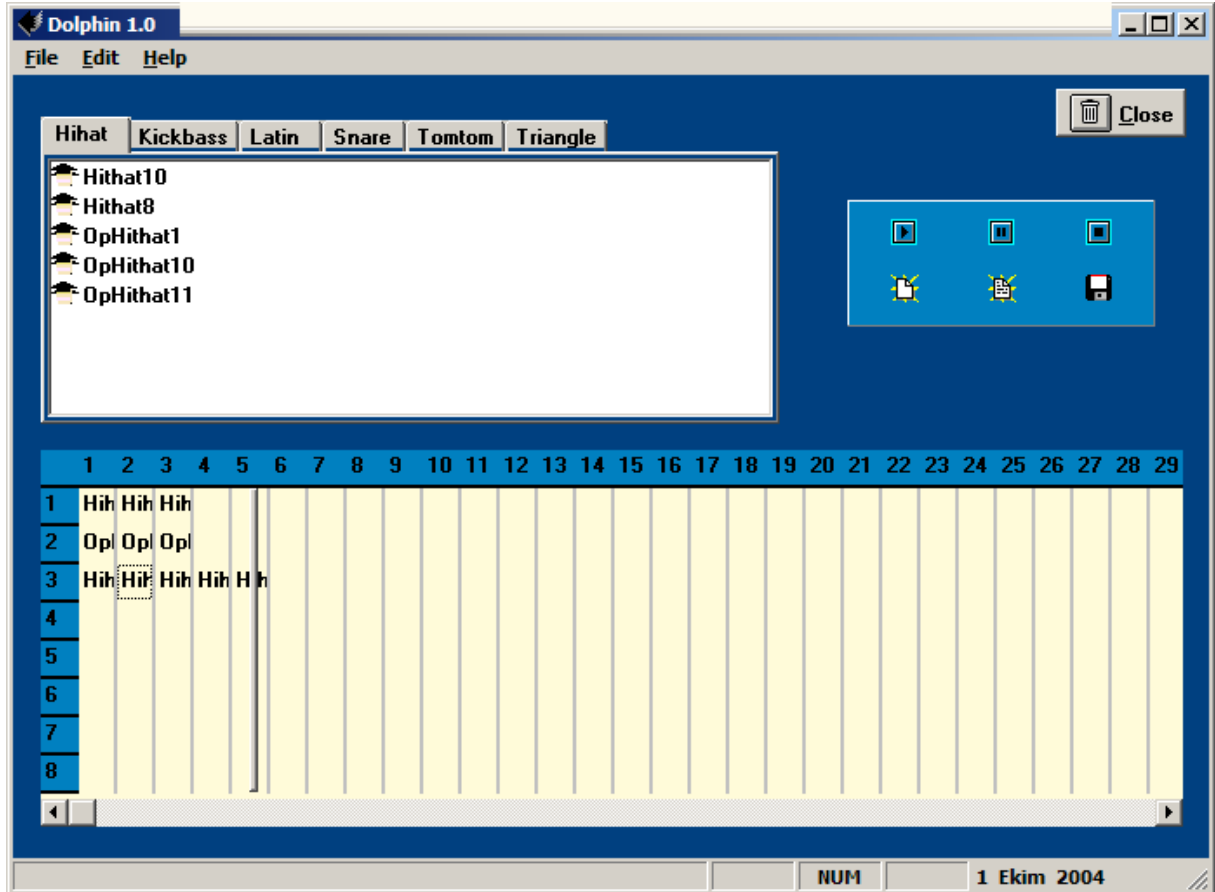
Öğretim elemanı raporu da ders, şubeler ve derslik bilgilerinden oluştuğundan tamamen derslane raporu gibi oluşturulur.

En son rapor olan Genel Ders Programı raporu ise okuldaki bölümler göz alınarak hazırlanır. Yani bir bölüme mensup olan tüm şubelerin ders, öğretim elemanı ve derslane verilerini içeren raporları ayrı ayrı hazırlanılarak birleştirilir ve bölümün genel ders programı haline getirilir.

24. Dolphin (müzik yapma programı)

(Ornekler\Delphi\Dolphin.zip)

Mayıs 2001 tarihli 2 hafta süreli bu proje aslında DanceEJay'den geliyor. WavMix nesnesini kullanarak 8 kanal müzik çalışıyoruz. Basit olarak Aç Kaydet diyalogları, durum çubuğu kullanımı, thread kullanımı ve şekilli form ve form sürüklenmesi yapıyoruz. Yardım dosyasını OasisSE ile yaptım. TListBox'dan TStringGrid nesnesine eleman sürükleyebiliyoruz. Müzik parçalarını Kopyala, Kes ve Yapıştır gibi işlemler varolmasına rağmen Geri Al kısmı yok. Ini dosyalama ile de *.dmf (Dolphin Music File) olarak kayıt edebiliyoruz.



Resim 33a – Ana pencere

Thread (kanal) programının kodları:

```
unit sesthread;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, WaveMix, Grids, ComCtrls, Buttons, ExtCtrls, StdCtrls;
```

```
type
```

```
sescal = class(TThread)
```

```
private
```

```
{ Private declarations }
```

```
filenm:string;
```

```
chan:integer;
```

```
w: array [0..10] of PMixWave;
```

```
protected
```

```
procedure Execute; override;
```

```

public
  constructor Create(dosya:string;kanal:integer);
end;

{ sescal }

implementation
uses kod;

constructor sescal.Create(dosya:string;kanal:integer);
begin
  filenm:=dosya;
  chan:=kanal;
  inherited Create(false);
end;

procedure sescal.Execute;
begin
  // repeat
  w[chan-1] := form1.WaveMix1.OpenFromFile(filenm);
  form1.WaveMix1.Play( chan-1, w[chan-1], nil, WMIX_QUEUEWAVE, 0 );
  // until Terminated;
end;
end.

```

Ana Programı:

```

unit kod;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  WaveMix, Grids, ComCtrls, Buttons, ExtCtrls,
  StdCtrls, sesthread, Menus,
  OleCtrls, IniFiles, ImgList;

type
  TForm1 = class(TForm)
    StringGrid1: TStringGrid;
    ImageList1: TImageList;
    Panel1: TPanel;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    OpenFileDialog1: TOpenDialog;
    SaveDialog1: TSaveDialog;
    Timer1: TTimer;
    SpeedButton5: TSpeedButton;
    PopupMenu1: TPopupMenu;
    Fillthesoundthisarea1: TMenuItem;
    Cleartheselectedarea1: TMenuItem;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    ListView1: TListView;
    TabSheet2: TTabSheet;
    TabSheet3: TTabSheet;
    TabSheet4: TTabSheet;
    TabSheet5: TTabSheet;
    TabSheet6: TTabSheet;
    ListView2: TListView;
    ListView3: TListView;
    ListView4: TListView;
    ListView5: TListView;
    ListView6: TListView;

```

```

    PopupMenu2: TPopupMenu;
    Play1: TMenuItem;
    StatusBar1: TStatusBar;
    BitBtn2: TBitBtn;
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    Open1: TMenuItem;
    Save1: TMenuItem;
    Clear1: TMenuItem;
    N1: TMenuItem;
    Close1: TMenuItem;
    Edit1: TMenuItem;
    Play2: TMenuItem;
    Help1: TMenuItem;
    About1: TMenuItem;
    Contents1: TMenuItem;
    Pause1: TMenuItem;
    Stop1: TMenuItem;
    N2: TMenuItem;
    Cleartheselectedarea2: TMenuItem;
    Fillthesoundthisarea2: TMenuItem;
    SpeedButton4: TSpeedButton;
    SpeedButton6: TSpeedButton;
    BitBtn1: TBitBtn;
    Insertxtimesto1: TMenuItem;
    Copy1: TMenuItem;
    Cut1: TMenuItem;
    Paste1: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    Copy2: TMenuItem;
    Cut2: TMenuItem;
    Paste2: TMenuItem;
    WaveMix1: TWaveMix;
    procedure FormCreate(Sender: TObject);
    procedure ListView1MouseDown(Sender: TObject; Button:
      TMouseButton;
      Shift: TShiftState; X, Y: Integer);

```

```

procedure StringGrid1DragDrop(Sender, Source:
Object; X, Y: Integer);
procedure StringGrid1DragOver(Sender, Source:
Object; X, Y: Integer;
    State: TDragState; var Accept: Boolean);
procedure Button1Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
procedure SpeedButton4Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton5Click(Sender: TObject);
procedure StringGrid1KeyDown(Sender: TObject; var
Key: Word;
    Shift: TShiftState);
procedure Fillthesoundthisarea1Click(Sender: TObject);
procedure Cleartheselectedarea1Click(Sender:
Object);
procedure ListView2MouseDown(Sender: TObject;
Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure ListView3MouseDown(Sender: TObject;
Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure ListView4MouseDown(Sender: TObject;
Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure ListView5MouseDown(Sender: TObject;
Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure ListView6MouseDown(Sender: TObject;
Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure Cal(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure Play1Click(Sender: TObject);
procedure FormMouseDown(Sender: TObject; Button:
TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure Timer1Timer(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure FormKeyDown(Sender: TObject; var Key:
Word;
    Shift: TShiftState);
procedure Close1Click(Sender: TObject);
procedure Contents1Click(Sender: TObject);
procedure Hinting(Sender: TObject); //ipuclari olayi
Procedure SetForm;
procedure FormResize(Sender: TObject);
procedure BitBtn1MouseDown(Sender: TObject; Button:
TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure FormDragOver(Sender, Source: TObject; X,
Y: Integer;
    State: TDragState; var Accept: Boolean);
procedure About1Click(Sender: TObject);
procedure SpeedButton6Click(Sender: TObject);
procedure Copy1Click(Sender: TObject);
procedure Cut1Click(Sender: TObject);
procedure Paste1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
    SesCalThread:array [1..8] of Sescal ;
end;

var

```

```

Form1: TForm1;
w: array [0..10] of PMixWave;
bellek: array [0..100, 0..8] of string;
var X,Y:Integer;
    FShow :Boolean;           // Mesaj için değişken
    F      :TForm;           // Aktif Olan Formu kullanmak için
    R      :HRgn;           // Formun Seklini belirtmek için
implementation

uses About;

{$R *.DFM}
procedure TForm1.Hinting(Sender: TObject); //ipuclari olayi
begin
    statusbar1.Panels[0].Text:=application.Hint;
end;

Procedure TForm1.SetForm;
var
    Dizi:Array[0..50] of TPoint;
    I      :Integer;
    L      :LongInt;
begin
    with form1 do begin
        X:=Width;
        Y:=Height;
        Dizi[0].X:=100;           Dizi[0].Y:=0;
        Dizi[1].X:=100;           Dizi[1].Y:=20;
        Dizi[2].X:=x-60;          Dizi[2].Y:=20;
        Dizi[3].X:=X-60;          Dizi[3].Y:=0;
        Dizi[4].X:=x;              Dizi[4].Y:=0;
        Dizi[5].X:=x;              Dizi[5].Y:=y;
        Dizi[6].X:=0;              Dizi[6].Y:=y;
        Dizi[7].X:=0;              Dizi[7].Y:=0;
        R:=CreatePolygonRgn(Dizi,8,Winding);
        SetWindowRgn( Handle,R,True);
    end;
    bitbtn2.Left:=x-96;
    Update;
end;

function GetFileSize(const FileName: string): LongInt;
var
    SearchRec: TSearchRec;
begin
    try
        if FindFirst(ExpandFileName(FileName), faAnyFile,
SearchRec) = 0 then
            Result := SearchRec.Size
        else Result := -1;
    finally
        SysUtils.FindClose(SearchRec);
    end;
end;

procedure TForm1.FormCreate(Sender: TObject);
var
    i : byte;
begin
    Application.OnHint:=Hinting;
    Application.HintColor:=claquaa;
    form1.SetForm;
    for i:=1 to 8 do stringgrid1.cells[0,i]:=inttostr(i);
    for i:=1 to 100 do begin

```



```

        stringgrid1.cells[i,0]:=inttostr(i);
    end;
end;

procedure TForm1.ListView1MouseDown(Sender:
TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var a:tlistitem;
begin
    if Button = mbLeft then
        begin
            a:=tlistitem.Create(listview1.Items);
            a:=listview1.GetItemAt(x,y);
            if a<>nil then listview1.BeginDrag(false);
        end;
    end;
end;

procedure TForm1.StringGrid1DragDrop(Sender, Source:
TObject; X,
Y: Integer);
var code,deger,cln,stn:integer;g:string;
dongu : byte;
clickedok:boolean;
begin
    if Source is Tbitbtn then
        begin
            bitbtn1.left:=x;
        end;
        stringgrid1.MouseToCell(x,y,cln,stn);
        if (stn<>0) and (cln<>0) then begin
            if Source is TListView then
                begin
                    initialize(deger);
                    g:='1';
                    clickedok:=InputQuery('Count', 'How many will you use
this sound?', g);
                    if clickedok then else exit;
                    Val(g, deger, Code);
                    stringgrid1.MouseToCell(x,y,cln,stn);
                    stringgrid1.row:=stn;
                    stringgrid1.col:=cln;
                    for dongu:=1 to deger do begin
                        if PageControl1.ActivePage.Caption='Hihat' then
                            if listview1.SelCount<>0 then
                                stringgrid1.cells[stringgrid1.col+(dongu-
1),stringgrid1.row] :=listview1.Selected.Caption;
                        if PageControl1.ActivePage.Caption='Kickbass' then
                            if listview2.SelCount<>0 then
                                stringgrid1.cells[stringgrid1.col+(dongu-
1),stringgrid1.row] :=listview2.Selected.Caption;
                        if PageControl1.ActivePage.Caption='Latin' then
                            if listview3.SelCount<>0 then
                                stringgrid1.cells[stringgrid1.col+(dongu-
1),stringgrid1.row] :=listview3.Selected.Caption;
                        if PageControl1.ActivePage.Caption='Snare' then
                            if listview4.SelCount<>0 then
                                stringgrid1.cells[stringgrid1.col+(dongu-
1),stringgrid1.row] :=listview4.Selected.Caption;
                        if PageControl1.ActivePage.Caption='Tomtom' then
                            if listview5.SelCount<>0 then
                                stringgrid1.cells[stringgrid1.col+(dongu-
1),stringgrid1.row] :=listview5.Selected.Caption;
                        if PageControl1.ActivePage.Caption='Triangle' then
                            if listview6.SelCount<>0 then
                                stringgrid1.cells[stringgrid1.col+(dongu-
1),stringgrid1.row] :=listview6.Selected.Caption;

```

```

        end; //for
    end;
end;
end;

procedure TForm1.StringGrid1DragOver(Sender, Source:
TObject; X,
Y: Integer; State: TDragState; var Accept: Boolean);
begin
    if sender is tlistitem then accept:=true;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
    close;
end;

procedure TForm1.SpeedButton3Click(Sender: TObject);
var dizin:string;
i,j:integer;
ayarini:TIniFile;
begin
    SaveDialog1.Execute;
    if Savedialog1.FileName<>'' then begin

        ayarini:=TIniFile.Create(Savedialog1.FileName);

        for i:=1 to 100 do
            for j:=1 to 8 do

                ayarini.WriteString(inttostr(i),inttostr(j),StringGrid1.cells[i,j]);

        ayarini.Free;
    end;
end;

procedure TForm1.SpeedButton4Click(Sender: TObject);
var i,j:integer;
ayarini:TIniFile;
begin
    OpenFileDialog1.Execute;
    if opendialog1.FileName<>'' then begin
        ayarini:=tinifile.create(opendialog1.FileName);
        for i:=1 to 100 do
            for j:=1 to 8 do

                StringGrid1.cells[i,j]:=ayarini.readstring(inttostr(i),inttostr(j),'');

        ayarini.Free;
    end;
end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
    WaveMix1.Activated:=false;
end;

procedure TForm1.SpeedButton5Click(Sender: TObject);
var i,j:integer;
begin
    if Application.MessageBox('Confirm to clear','Question',1)
=IDOK then
        begin
            cursor:=crHourGlass;
            for i:=1 to 100 do

```

```

    for j:=1 to 8 do
        StringGrid1.Cells[i,j]:="";
    cursor:=crDefault;
    end;
end;

procedure TForm1.StringGrid1KeyDown(Sender: TObject;
var Key: Word;
Shift: TShiftState);
var i,j :integer;
begin
if key = VK_DELETE then
    for j:=1 to StringGrid1.Selection.bottom-
StringGrid1.Selection.top+1 do
        for i:=1 to StringGrid1.Selection.Right-
StringGrid1.Selection.Left+1 do
            begin
                stringgrid1.cells[StringGrid1.Selection.Left+i-
1,StringGrid1.Selection.Top+j-1]:="";
            end;
        end;
    end;

procedure TForm1.Fillthesoundthisarea1Click(Sender:
TObject);
var i,j :integer;
begin
    for j:=1 to StringGrid1.Selection.bottom-
StringGrid1.Selection.top+1 do
        for i:=1 to StringGrid1.Selection.Right-
StringGrid1.Selection.Left+1 do
            begin
                if PageControl1.ActivePage.Caption='Hihat' then
                    if listview1.SelCount<>0 then
                        stringgrid1.cells[StringGrid1.Selection.Left+i-
1,StringGrid1.Selection.Top+j-1]
:=listview1.Selected.Caption;
                    if PageControl1.ActivePage.Caption='Kickbass' then
                        if listview2.SelCount<>0 then
                            stringgrid1.cells[StringGrid1.Selection.Left+i-
1,StringGrid1.Selection.Top+j-1]
:=listview2.Selected.Caption;
                        if PageControl1.ActivePage.Caption='Latin' then
                            if listview3.SelCount<>0 then
                                stringgrid1.cells[StringGrid1.Selection.Left+i-
1,StringGrid1.Selection.Top+j-1]
:=listview3.Selected.Caption;
                        if PageControl1.ActivePage.Caption='Snare' then
                            if listview4.SelCount<>0 then
                                stringgrid1.cells[StringGrid1.Selection.Left+i-
1,StringGrid1.Selection.Top+j-1]
:=listview4.Selected.Caption;
                        if PageControl1.ActivePage.Caption='Tomtom' then
                            if listview5.SelCount<>0 then
                                stringgrid1.cells[StringGrid1.Selection.Left+i-
1,StringGrid1.Selection.Top+j-1]
:=listview5.Selected.Caption;
                        if PageControl1.ActivePage.Caption='Triangle' then
                            if listview6.SelCount<>0 then
                                stringgrid1.cells[StringGrid1.Selection.Left+i-
1,StringGrid1.Selection.Top+j-1]
:=listview6.Selected.Caption;
                    end;
                end;
            end;
end;
end;

```

```

procedure TForm1.Cleartheselectedarea1Click(Sender:
TObject);
var i,j ,fark:integer;
begin
    for j:=1 to StringGrid1.Selection.bottom-
StringGrid1.Selection.top+1 do
        for i:=1 to StringGrid1.Selection.Right-
StringGrid1.Selection.Left+1 do
            begin
                stringgrid1.cells[StringGrid1.Selection.Left+i-
1,StringGrid1.Selection.Top+j-1]:="";
            end;
        end;
    end;

procedure TForm1.ListView2MouseDown(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var a:tlistitem;
begin
    if Button = mbLeft then
        begin
            a:=tlistitem.Create(listview2.Items);
            a:=listview2.GetItemAt(x,y);
            if a<>nil then listview2.BeginDrag(false);
        end;
    end;

end;

procedure TForm1.ListView3MouseDown(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var a:tlistitem;
begin
    if Button = mbLeft then
        begin
            a:=tlistitem.Create(listview3.Items);
            a:=listview3.GetItemAt(x,y);
            if a<>nil then listview3.BeginDrag(false);
        end;
    end;

end;

procedure TForm1.ListView4MouseDown(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var a:tlistitem;
begin
    if Button = mbLeft then
        begin
            a:=tlistitem.Create(listview4.Items);
            a:=listview4.GetItemAt(x,y);
            if a<>nil then listview4.BeginDrag(false);
        end;
    end;

end;

procedure TForm1.ListView5MouseDown(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var a:tlistitem;
begin
    if Button = mbLeft then
        begin
            a:=tlistitem.Create(listview5.Items);
            a:=listview5.GetItemAt(x,y);
            if a<>nil then listview5.BeginDrag(false);
        end;
    end;

end;

```

end;

```
procedure TForm1.ListView6MouseDown(Sender:
TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var a:tlistitem;
begin
if Button = mbLeft then
begin
a:=tlistitem.Create(listview6.Items);
a:=listview6.GetItemAt(x,y);
if a<>nil then listview6.BeginDrag(false);
end;
end;
end;
```

```
procedure TForm1.Cal(Sender: TObject);
var i,j:integer;dosya:string;
devam:boolean;sayac:integer;
begin
try
WaveMix1.Activated:=true;
for i:=1 to 100 do begin
sayac:=0;
for j:=1 to 8 do begin
if stringgrid1.cells[i,j]=' ' then inc(sayac);
if stringgrid1.cells[i,j]='Hihat10' then begin
dosya:='Hihat\hihat10.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='Hihat8' then begin
dosya:='Hihat\hihat8.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='OpHihat1' then begin
dosya:='Hihat\ophihat1.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='OpHihat10' then begin
dosya:='Hihat\ophihat10.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='OpHihat11' then begin
dosya:='Hihat\ophihat11.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='Hihat10' then begin
dosya:='Hihat\hihat10.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;

if stringgrid1.cells[i,j]='Technokick3' then begin
dosya:='Kickbass\technokick3.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='Technokick2' then begin
dosya:='Kickbass\technokick2.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='Kickbass2' then begin
dosya:='Kickbass\kickbass2.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='Kickbass1' then begin
dosya:='Kickbass\kickbass1.wav';
```

```
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='Technokick1' then begin
dosya:='Kickbass\technokick1.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
```

```
if stringgrid1.cells[i,j]='Woodblock1' then begin
dosya:='Latin\woodblock1.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
```

```
if stringgrid1.cells[i,j]='Snare1' then begin
dosya:='Snare\snare1.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='Synthsnare2' then begin
dosya:='Snare\synthsnare2.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='Rocksnare1' then begin
dosya:='Snare\rocksnare1.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
```

```
if stringgrid1.cells[i,j]='Tom1' then begin
dosya:='Tomtom\tom1.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='Tom2' then begin
dosya:='Tomtom\tom2.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='Tom3' then begin
dosya:='Tomtom\tom3.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
```

```
if stringgrid1.cells[i,j]='Triangle1' then begin
dosya:='Triangle\triangle1.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='Triangle2' then begin
dosya:='Triangle\triangle3.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
if stringgrid1.cells[i,j]='Triangle3' then begin
dosya:='Triangle\triangle3.wav';
SesCalThread[1]:=sescal.Create(dosya,j);
end;
end;
if sayac=8 then exit
else bitbtn1.left:=bitbtn1.left+20;
```

```
end;
except
Application.MessageBox('Your sound card is
busy!','Error',0);
end;
end;
```

```
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
bitbtn1.left:=36;
// WaveMix1.FlushAllChannels(True);
form1.Cal(sender);
```

```

end;

procedure TForm1.Play1Click(Sender: TObject);
var dosya : string;j:integer;
begin
j:=1;
WaveMix1.Activated:=true;
if PageControl1.ActivePage.Caption='Hihat' then begin
if ListView1.SelCount<>0 then begin
if listview1.Selected.Caption='Hihat10' then begin
dosya:='Hihat\hihat10.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
if listview1.Selected.Caption='Hihat8' then begin
dosya:='Hihat\hihat8.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
if listview1.Selected.Caption='OpHihat1' then begin
dosya:='Hihat\lophihat1.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
if listview1.Selected.Caption='OpHihat10' then begin
dosya:='Hihat\lophihat10.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
if listview1.Selected.Caption='OpHihat11' then begin
dosya:='Hihat\lophihat11.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
end;
end;
if PageControl1.ActivePage.Caption='Kickbass' then begin
if ListView2.SelCount<>0 then begin
if listview2.Selected.Caption='Technokick3' then begin
dosya:='Kickbass\technokick3.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
if listview2.Selected.Caption='Technokick2' then begin
dosya:='Kickbass\technokick2.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
if listview2.Selected.Caption='Kickbass2' then begin
dosya:='Kickbass\kickbass2.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
if listview2.Selected.Caption='Kickbass1' then begin
dosya:='Kickbass\kickbass1.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
if listview2.Selected.Caption='Technokick1' then begin
dosya:='Kickbass\technokick1.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
end;
end;
if PageControl1.ActivePage.Caption='Latin' then
if ListView3.SelCount<>0 then
if listview3.Selected.Caption='Woodblock1' then begin
dosya:='Latin\woodblock1.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
end;
if PageControl1.ActivePage.Caption='Snare' then begin
if ListView4.SelCount<>0 then begin
if listview4.Selected.Caption='Snare1' then begin
dosya:='Snare\snare1.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
if listview4.Selected.Caption='Synthsnare2' then begin
dosya:='Snare\synthsnare2.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
if listview4.Selected.Caption='Rocksnare1' then begin
dosya:='Snare\rocksnare1.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
end;
end;
if PageControl1.ActivePage.Caption='Tomtom' then begin
if ListView5.SelCount<>0 then begin
if listview5.Selected.Caption='Tom1' then begin
dosya:='Tomtom\tom1.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
if listview5.Selected.Caption='Tom2' then begin
dosya:='Tomtom\tom2.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
if listview5.Selected.Caption='Tom3' then begin
dosya:='Tomtom\tom3.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
end;
end;
if PageControl1.ActivePage.Caption='Triangle' then begin
if ListView6.SelCount<>0 then begin
if listview6.Selected.Caption='Triangle1' then begin
dosya:='Triangle\triangle1.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
if listview6.Selected.Caption='Triangle2' then begin
dosya:='Triangle\triangle3.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
if listview6.Selected.Caption='Triangle3' then begin
dosya:='Triangle\triangle3.wav';
SesCalThread[2]:=sescal.Create(dosya,j);
end;
end;
end;
end;

procedure TForm1.FormMouseDown(Sender: TObject; Button:
TMouseButton;
Shift: TShiftState; X, Y: Integer);
const
SC_DragMove = $F012;
begin
Begin
ReleaseCapture;
FORM1.perform(WM_SysCommand, SC_DragMove, 0);
End;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
const
CapPanel = 1;
NumPanel = 2;
Scr1Panel = 3;
DatePanel = 4;
begin

```

```

with StatusBar1 do
begin
  if GetKeyState(VK_CAPITAL) <> 0 then
    StatusBar1.Panels[CapPanel].Text := 'CAP'
  else
    StatusBar1.Panels[CapPanel].Text := '';
  if GetKeyState(VK_NUMLOCK) <> 0 then
    StatusBar1.Panels[NumPanel].Text := 'NUM'
  else
    StatusBar1.Panels[NumPanel].Text := '';
  if GetKeyState(VK_SCROLL) <> 0 then
    StatusBar1.Panels[ScrlPanel].Text := 'SCRL'
  else
    StatusBar1.Panels[ScrlPanel].Text := '';
  StatusBar1.Panels[DatePanel].Text :=
    FormatDateTime('D mmmm YYYY', now);
end;
end;

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
close;
end;

procedure TForm1.FormKeyDown(Sender: TObject; var
Key: Word;
Shift: TShiftState);
begin
  if key=VK_F1 then
    Application.HelpCommand(HELP_CONTENTS, 0);
end;

procedure TForm1.Close1Click(Sender: TObject);
begin
close;
end;

procedure TForm1.Contents1Click(Sender: TObject);
begin
  Application.HelpCommand(HELP_CONTENTS, 0);
end;

procedure TForm1.FormResize(Sender: TObject);
begin
Form1.setform;
end;

procedure TForm1.BitBtn1MouseDown(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var a:TBitBtn;
begin
  if Button = mbLeft then
    begin
      a:=TBitBtn.Create(BitBtn1);
      // a:=bitbtn1.GetItemAt(x,y);
      if a<>nil then bitbtn1.BeginDrag(false);
    end;
end;

end;

procedure TForm1.FormDragOver(Sender, Source:
TObject; X, Y: Integer;
State: TDragState; var Accept: Boolean);
begin
  if sender is tbitbtn then

```

```

accept:=true;
end;

procedure TForm1.About1Click(Sender: TObject);
begin
aboutbox.show;
end;

procedure TForm1.SpeedButton6Click(Sender: TObject);
begin
WaveMix1.FlushAllChannels(True);
end;

procedure TForm1.Copy1Click(Sender: TObject);
var i,j:integer;
begin
  for i:=0 to 99 do for j:=0 to 8 do bellek [i,j]:= '';

  for j:=1 to StringGrid1.Selection.bottom-
StringGrid1.Selection.top+1 do
    for i:=1 to StringGrid1.Selection.Right-
StringGrid1.Selection.Left+1 do
      begin
        bellek [i-1,j-
1]:=stringgrid1.cells[StringGrid1.Selection.Left+i-
1,StringGrid1.Selection.Top+j-1];
      end;
      bellek [i-1,0]:='/' //end
    end;

  procedure TForm1.Cut1Click(Sender: TObject);
  var i,j:integer;
  begin
    for i:=0 to 99 do for j:=0 to 8 do bellek [i,j]:= '';

    for j:=1 to StringGrid1.Selection.bottom-
StringGrid1.Selection.top+1 do
      for i:=1 to StringGrid1.Selection.Right-
StringGrid1.Selection.Left+1 do
        begin
          bellek [i-1,j-
1]:=stringgrid1.cells[StringGrid1.Selection.Left+i-
1,StringGrid1.Selection.Top+j-1];
        end;
        bellek [i-1,0]:='/' //end
      end;

      for j:=1 to StringGrid1.Selection.bottom-
StringGrid1.Selection.top+1 do
        for i:=1 to StringGrid1.Selection.Right-
StringGrid1.Selection.Left+1 do
          begin
            stringgrid1.cells[StringGrid1.Selection.Left+i-
1,StringGrid1.Selection.Top+j-1]:= '';
          end;
        end;

  procedure TForm1.Paste1Click(Sender: TObject);
  var i,j:integer;sayacx,sayacy,son:integer;
  begin

    for i:=0 to 99 do begin
      if bellek [i,0]='/' then son:=i;
    end;
    sayacx:=0;
    for i:=StringGrid1.Selection.left to
StringGrid1.Selection.left+son-1 do begin

```

```

sayacy:=0;
for j:=StringGrid1.Selection.Top to 8 do begin
  if bellek [sayacx,sayacy]<>" then
    stringgrid1.Cells[i,j]:=bellek [sayacx,sayacy] ;
  inc(sayacy);
end;

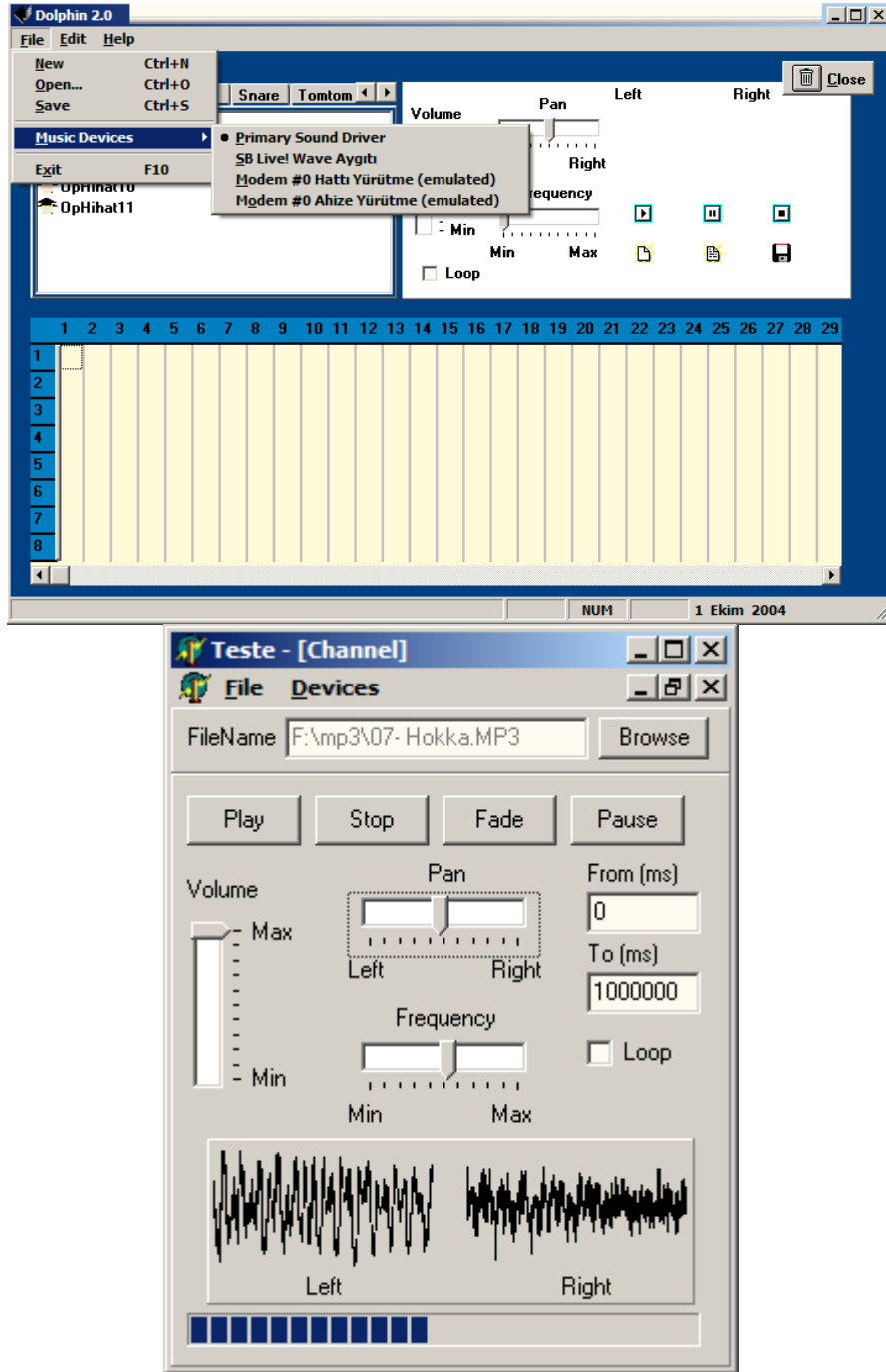
```

```

inc(sayacx);
end;
end;

end.

```



Resim 33b – Dolphin2 ekranı ve DSMIX

Daha geliştirdiğim Dophin2 de ise, DSMixer v1.51 yani DirectX'in DirectSound komutları ile kaliteli müzik çalabiliyoruz. Maalesef DSMIX klasöründeki örneği Dolphin'e tam uyarlayamadım. Eh, yine size iş çıktı!

25. Ofis (basit bir veritabanı uygulaması)

(Ornekler\Delphi\Ofis.zip)

Mart 2001 tarihli 1500 satırdan oluşan bir işyeri için askerde yaptığım program. Sade bir şekilde faturalama, veri girişi ve listeleme yöntemlerimiz var. Bu program ile ilk kez yaptığım programcılıktan para kazanacağımı ummuştum, fakat olmadı. Neyse, önemli değil...



Resim 34a – Giriş ve hakkında pencereleri

Çok az programlama yaparak küçük bir işyerinin ihtiyacı olan benzin satışı ve eski depo kayıtlarına ulaşmayı sağlıyoruz. Delphi'nin temel veritabanı yetenekleri sayesinde neredeyse hiç program yazmadık. Örneğin bol bol hesaplanan alanlar kullandık.

Her gece ve her gündüz zamanlarında, güvenilir bir eleman stoktan ne kadar ürün satıldığını pompalardan bakarak işaretler, bu sıfırlamayı kağıda da dökülebilir, tabi bu arada müşterilerin aldığı malzemeler de fatura edilerek dökümü alınır. Asıl istenen ilk işar ile son işar arasındaki farkı bulup sıfırlama işleminin yapılması. Eğer aktarma yani sıfırlama sırasında kar elde edilmişse çalışanlara da istenen miktarda para eşit olarak bölünebiliyor. Kötü bir fikir olarak da, açık var olduğu düşünülmüş ise personelden para da kesilebilir. Yani açık ve avans miktarı da personel veritabanında saklanıyor.

Bence bu programın altyapısı kullanılarak istenen türde küçük ve orta boy yerel veritabanı uygulamaları yapılabilir. Tek eksiği kullanıcı yetkilendirmesi yani kullanıcı girişi olmamasından dolayı, bilgisayarı açık bulan bir çalışan istediği bilgiyi değiştirebilmesi.

Personel Bilgileri

Personel No

Adı Soyadı Sicil No Resim

Doğum Tarihi İşe Giriş Tarihi

Adres Telefon

Ağırlık Toplam

Avans

Resim Ekle
✓ Sil

Satış Raporu

Kayıt Tarihi **01.01.2002** Gece Bugünün Tarihi **01.10.2004**

Sıfırla Kapat

	Süper	Normal	Motorin	Kurşunsuz	Gaz
Son İşar					
İlk İşar	1500	1600	1700	1800	1900
Satılan	-1500	-1600	-1700	-1800	-1900
Fiyat	600.500	600.450	251.000	350.000	300.300
Para (TL)	-900750000	-960720000	-426700000	-630000000	-570570000

10 Milyon 5 X

5 Milyon 0

1 Milyon 5

500 Bin 0

250 Bin 0

Bozukluk 0

KDV Oranı % 15

KDV -523311000

Yağ -3488740000

Ara Toplam -3488740000

Nakit 55.000.000

Kredi

Hesap= -3543740000

Nakite Aktar

Resim 34b – Satış ve personel kayıt formu

Personel Bilgileri

Adı Soyadı Adres Telefon

Vergi Dairesi Vergi Numarası

Satış Raporu

Kayıt Tarihi **01.01.2002** Gece Bugünün Tarihi **01.10.2004**

Sıfırla Kapat

	Süper	Normal	Motorin	Kurşunsuz	Gaz
Son İşar					
İlk İşar	1500	1600	1700	1800	1900
Satılan	-1500	-1600	-1700	-1800	-1900
Fiyat	600.500	600.450	251.000	350.000	300.300
Para (TL)	-900750000	-960720000	-426700000	-630000000	-570570000

10 Milyon 5 X

5 Milyon 0

1 Milyon 5

500 Bin 0

250 Bin 0

Bozukluk 0

KDV Oranı % 15

KDV -523311000

Yağ -3488740000

Ara Toplam -3488740000

Nakit 55.000.000

Kredi

Hesap= -3543740000

Nakite Aktar

Faturalama Penceresi

Adı Soyadı Adres Telefon

Vergi Dairesi Vergi Numarası

Satış Raporu

Kayıt Tarihi **01.01.2002** Gece Bugünün Tarihi **01.10.2004**

Sıfırla Kapat

	Süper	Normal	Motorin	Kurşunsuz	Gaz
Son İşar					
İlk İşar	1500	1600	1700	1800	1900
Satılan	-1500	-1600	-1700	-1800	-1900
Fiyat	600.500	600.450	251.000	350.000	300.300
Para (TL)	-900750000	-960720000	-426700000	-630000000	-570570000

10 Milyon 5 X

5 Milyon 0

1 Milyon 5

500 Bin 0

250 Bin 0

Bozukluk 0

KDV Oranı % 15

KDV -523311000

Yağ -3488740000

Ara Toplam -3488740000

Nakit 55.000.000

Kredi

Hesap= -3543740000

Nakite Aktar

Resim 34c – Faturalama pencereleri

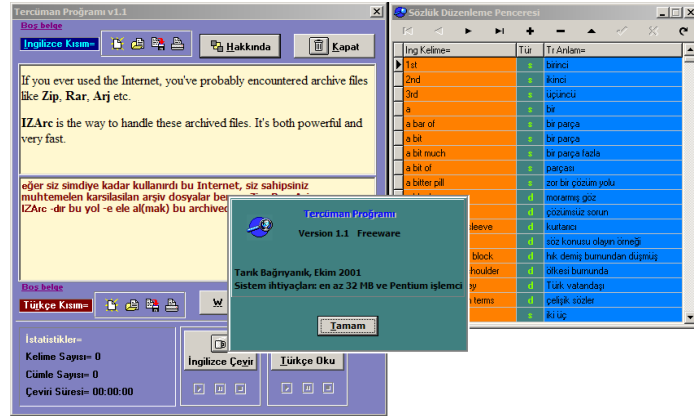
26. Tercuman (İngilizce'den Türkçe'ye çeviri ve okuma programı)

(Ornekler\Delphi\Tercuman.zip)

Ekim 2001 tarihli, yaklaşık 1000 satırdan oluşan bu programda basit bir sözlük, ona bağlı olarak aynen (gramer olarak değil yani) İngilizce'den Türkçe'ye hızlı bir şekilde çeviriyor ve okumaya da çalışıyoruz. DXWaveList ve DXSound nesneleri ile temiz ses çalışı elde etmeye çalışıyoruz. Çeviri ve okuma kısımları thread ile yapıldığı için istediğimiz zaman ara verip, sonra devam edebiliyoruz veya tamamen işlemi bitirebiliyoruz. Sözlüğü başka bir programdan almıştım, bu sözlük üzerinde düzeltme ve silme gibi işlemler rahatlıkla yapılabilir.

İlk versiyonda gramer için çok uğraşırken, daha sonra seslendirmeyi eklemek isteyince buna pek zaman ayıramadım. Yani cümleyi çevirmeye değil, şimdilik kelime kelime karşılık yazarak idare ediyoruz. Ama İngilizce çeviri, Türkçe'yi başka bir dile çevirmekten kolay. Çünkü Türkçe'de fiiller ve cümledeki kelimelerin yerine göre anlam değişmesi çok fazla. Sonuç olarak çevrilen kısmı farklı kaydedebilir, yazıcıdan da çıkarabiliriz. Sonraki sürümlerde site , eposta çeviri ve okuması eklenebilir. Cümleleri "." işareti ile bitirmek gerekli.

Dolphin müzik programı gibi DXSound nesnesini kullanıyoruz. Wavmix'de işe yarayabilir, fakat daha önce de dediğim gibi DirectSound daha kaliteli. Kendi sesimi a, b, c şeklinde çalışıyoruz. Bazı harfleri tek başlarına söylemeye çalışmak yanlış. Mesela k harfi, ke, kü, kö, ko, ki, k gibi birçok şekilde okunabiliyor. Ses birleştirmeyi, sentezlemeyi başaramadım. Çok uzun metinlerde hata veriyor ve program kilitleniyor. Bunun sebebi hatalı thread programlama da olabilir. Kısa kelimeler ile okuma kısmını deneyebilirsiniz.



Resim 35 – Tercüman Programı

Ana kod:

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics,  
Controls, Forms, Dialogs, ExtCtrls, DBCtrls, Grids,  
DBGrids, Db, DBTables, StdCtrls, Buttons, Unit2, unit3,  
ComCtrls, WaveMix, Menus, DXSounds, Wave;
```

```
type
```

```
TForm1 = class(TForm)  
    SpeedButton1: TSpeedButton;  
    BitBtn1: TBitBtn;  
    StaticText1: TStaticText;  
    StaticText2: TStaticText;  
    SpeedButton2: TSpeedButton;  
    SpeedButton3: TSpeedButton;  
    SpeedButton4: TSpeedButton;  
    SpeedButton5: TSpeedButton;  
    SpeedButton6: TSpeedButton;
```

```
SpeedButton7: TSpeedButton;  
OpenDialog1: TOpenDialog;  
SaveDialog1: TSaveDialog;  
Bevel2: TBevel;  
Label1: TLabel;  
Label2: TLabel;  
Label3: TLabel;  
Label5: TLabel;  
SpeedButton8: TSpeedButton;  
SpeedButton9: TSpeedButton;  
SpeedButton10: TSpeedButton;  
Label4: TLabel;  
Label6: TLabel;  
Bevel3: TBevel;  
Bevel4: TBevel;  
BitBtn2: TBitBtn;  
SpeedButton11: TSpeedButton;  
SpeedButton12: TSpeedButton;  
Bevel5: TBevel;  
SpeedButton13: TSpeedButton;  
Query1: TQuery;
```

```

Query1KELIME: TStringField;
Query1TUR: TStringField;
Query1ANLAM: TStringField;
Panel1: TPanel;
RichEdit1: TRichEdit;
RichEdit2: TRichEdit;
Splitter1: TSplitter;
SpeedButton14: TSpeedButton;
PopupMenu1: TPopupMenu;
GeriAL1: TMenuItem;
Kopyala1: TMenuItem;
Kes1: TMenuItem;
Yaptr1: TMenuItem;
N1: TMenuItem;
PopupMenu2: TPopupMenu;
MenuItem1: TMenuItem;
MenuItem2: TMenuItem;
MenuItem3: TMenuItem;
MenuItem4: TMenuItem;
MenuItem5: TMenuItem;
Bevel1: TBevel;
DXWaveList1: TDXWaveList;
DXSound1: TDXSound;
Bevel6: TBevel;
SpeedButton15: TSpeedButton;
SpeedButton16: TSpeedButton;
SpeedButton17: TSpeedButton;
procedure BitBtn1Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton5Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
procedure SpeedButton6Click(Sender: TObject);
procedure SpeedButton4Click(Sender: TObject);
procedure SpeedButtonClick(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton8Click(Sender: TObject);
procedure SpeedButton9Click(Sender: TObject);
procedure SpeedButton10Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure SpeedButton11Click(Sender: TObject);
procedure SpeedButton12Click(Sender: TObject);
procedure SpeedButton13Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure SpeedButton14Click(Sender: TObject);
procedure Yaptr1Click(Sender: TObject);
procedure Kes1Click(Sender: TObject);
procedure Kopyala1Click(Sender: TObject);
procedure GeriAL1Click(Sender: TObject);
procedure MenuItem1Click(Sender: TObject);
procedure MenuItem3Click(Sender: TObject);
procedure MenuItem4Click(Sender: TObject);
procedure MenuItem5Click(Sender: TObject);
procedure SpeedButton17Click(Sender: TObject);
procedure SpeedButton16Click(Sender: TObject);
procedure SpeedButton15Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
cvr:cevir;
sees:ses;
w: array [0..30] of PMixWave;
end;

var
Form1: TForm1;

```

```

kelime_Sayisi,cumle_sayisi,ceviri_suresi:integer;
devam:boolean;
ilkzaman:TDateTime;
klasor:string;

const
harfler: array [1..28] of string = ('a','b','c','ç','d','e','f','g','h','ı',
'i','j','k','l','m','n','o','ö','p','r','s','ş','t','u','ü','v','y','z');

implementation

uses unit4, sozluk;

{$R *.DFM}

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
close;
end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
richedit1.Lines.Clear;
label4.caption:='Boş belge';
end;

procedure TForm1.SpeedButton5Click(Sender: TObject);
begin
richedit2.Lines.Clear;
label6.caption:='Boş belge';
end;

procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
OpenDialog1.Execute;
if OpenDialog1.FileName<>'' then begin
richedit1.Lines.LoadFromFile(OpenDialog1.FileName);

label4.Caption:=OpenDialog1.FileName;
end;
end;

procedure TForm1.SpeedButton6Click(Sender: TObject);
begin
OpenDialog1.Execute;
if OpenDialog1.FileName<>'' then begin
richedit2.Lines.LoadFromFile(OpenDialog1.FileName);
label6.Caption:=OpenDialog1.FileName;
end;
end;

procedure TForm1.SpeedButton4Click(Sender: TObject);
begin
SaveDialog1.Execute;
if SaveDialog1.FileName<>'' then
richedit1.Lines.SaveToFile(SaveDialog1.FileName);
end;

procedure TForm1.SpeedButtonClick(Sender: TObject);
begin
SaveDialog1.Execute;
if SaveDialog1.FileName<>'' then
richedit2.Lines.SaveToFile(SaveDialog1.FileName);
end;

```

```

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
query1.Active:=true;
label2.Caption:='Kelime Sayısı= 0 ';
label3.Caption:='Cümle Sayısı= 0 ';
label5.Caption:='Çeviri Süresi= 00:00:00';
kelime_Sayisi:=0;cumle_sayisi:=0;ceviri_suresi:=0;
ilkzaman:=now();

cvr:=cevir.Create(False);
cvr.Priority:=tpNormal;

SpeedButton1.Enabled:=false;
SpeedButton8.Enabled:=true;
SpeedButton9.Enabled:=true;
SpeedButton10.Enabled:=true;
end;

procedure TForm1.SpeedButton10Click(Sender: TObject);
begin
cvr.Suspend;
cvr.free;

form1.SpeedButton1.Enabled:=true;
form1.SpeedButton8.Enabled:=false;
form1.SpeedButton9.Enabled:=false;
form1.SpeedButton10.Enabled:=false;
form1.SpeedButton9.Down:=true;
end;

procedure TForm1.SpeedButton8Click(Sender: TObject);
begin
cvr.Suspend;
end;

procedure TForm1.SpeedButton9Click(Sender: TObject);
begin
cvr.Resume;
end;

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
aboutbox.show;
end;

procedure TForm1.SpeedButton11Click(Sender: TObject);
begin
if Application.MessageBox('Yazdırma
Onayı','Yazdır',1)=idOK then
RichEdit1.print('Tercüman Döküm');
end;

procedure TForm1.SpeedButton12Click(Sender: TObject);
begin
if Application.MessageBox('Yazdırma
Onayı','Yazdır',1)=idOK then
RichEdit2.print('Tercüman Döküm');
end;

procedure TForm1.SpeedButton13Click(Sender: TObject);
//seslendirme
var i,j,k,l:integer;a:string;

```

```

begin
sees:=ses.Create(False);
sees.Priority:=tpHigher;
form1.SpeedButton13.Enabled:=false;
SpeedButton13.Enabled:=false;
SpeedButton17.Enabled:=true;
SpeedButton16.Enabled:=true;
SpeedButton15.Enabled:=true;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
klasor:='takozl';
end;

procedure TForm1.SpeedButton14Click(Sender: TObject);
begin
form5.show;
end;

procedure TForm1.Yaptr1Click(Sender: TObject);
begin
richedit1.PasteFromClipboard;
end;

procedure TForm1.Kes1Click(Sender: TObject);
begin
richedit1.CutToClipboard;
end;

procedure TForm1.Kopyala1Click(Sender: TObject);
begin
richedit1.CopyToClipboard;
end;

procedure TForm1.GeriAl1Click(Sender: TObject);
begin
richedit1.Undo;
end;

procedure TForm1.Menuitem1Click(Sender: TObject);
begin
richedit2.Undo;
end;

procedure TForm1.Menuitem3Click(Sender: TObject);
begin
richedit2.CopyToClipboard;
end;

procedure TForm1.Menuitem4Click(Sender: TObject);
begin
richedit2.CutToClipboard;
end;

procedure TForm1.Menuitem5Click(Sender: TObject);
begin
richedit2.PasteFromClipboard;
end;

procedure TForm1.SpeedButton17Click(Sender: TObject);
begin
sees.Resume;
end;

procedure TForm1.SpeedButton16Click(Sender: TObject);

```

```

begin
sees.suspend;
end;

procedure TForm1.SpeedButton15Click(Sender: TObject);
begin
    sees.Suspend;
    sees.free;

```

Çeviri thread - kanal kodu:

```

unit Unit2;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics,
    Controls, Forms, Dialogs,
    ExtCtrls, DBCtrls, Grids, DBGrids, Db, DBTables,
    StdCtrls, Buttons;

type
    cevir = class(TThread)
    private
        { Private declarations }
    protected
        procedure Execute; override;
        procedure translate;
    end;

    var cumle:array[1..50]of string;
        isimler:array[1..50]of string;
        fiil:array[1..50]of string;
        diger:array[1..50]of string;

implementation

{ cevir }

uses unit1;

procedure cevir.translate;

var i,j,k,l,harf:integer;a,b:string;boyut_a:integer;
    kelimeler,ara,turkcesatir:string;ilk_kelime:boolean;
    isim_sayisi,fiil_sayisi,diger_sayisi:integer;
    bulunanin_tipi:string;noktadan_onceki_k:integer;
label sart2;
begin
    form1.richedit2.Clear;
    j:=form1.richedit1.Lines.Capacity;
    isim_sayisi:=0; //cumledeki isim kac tane
    fiil_sayisi:=0;
    diger_sayisi:=0;

    for i:=0 to j do begin //satir sayisi
        a:=trimright((pchar(form1.RichEdit1.Lines.Strings[i])));
        //satir stringi
        harf:=1;kelimeler:='';b:='';
        turkcesatir:='';ilk_kelime:=true;
        boyut_a:=length(a);
        noktadan_onceki_k:=0;
        for k:=1 to boyut_a do begin //harfler

```

```

SpeedButton13.Enabled:=true;
SpeedButton15.Enabled:=false;
SpeedButton16.Enabled:=false;
SpeedButton17.Enabled:=false;
SpeedButton17.Down:=true;
end;

end.

```

```

if (a[k]='.')and(a[k-1]<>'.') then begin //yeni cumleye gec...

```

```

    kelimeler:='';
    for l:= noktadan_onceki_k to k-1 do kelimeler[l]:=a[l];
    ////////////
    with form1.query1 do
    begin
        ara:=trim(kelimeler);

```

```

DisableControls;close;params[0].AsString:=strlower(pchar(ara)
);

```

```

    open;enablecontrols;
end;

```

```

    b:=form1.query1Anlam.value;
    bulunanin_tipi:=form1.query1Tur.value;

```

```

    if b="" then b:=kelimeler; //bulamazsa aynen yazar

```

```

    if bulunanin_tipi='i' then
    begin
        inc(isim_sayisi);
        isimler[isim_sayisi]:=b;
    end
    else
    if bulunanin_tipi='y' then
    begin
        inc(fiil_sayisi);
        fiil[fiil_sayisi]:=b;
    end
    else
    begin
        inc(diger_sayisi);
        diger[diger_sayisi]:=b;
    end;

```

```

    ////////////
    for l:=1 to diger_sayisi do
    begin
        turkcesatir:=trim(turkcesatir+' '+diger[l]);
    end;
    for l:=1 to isim_sayisi do
    begin
        turkcesatir:=trim(turkcesatir+' '+isimler[l]);
    end;
    for l:=1 to fiil_sayisi do
    begin
        turkcesatir:=trim(turkcesatir+' '+fiil[l]);
    end;

```

```

    form1.RichEdit2.lines.Append(turkcesatir+'.');
    turkcesatir:='';

```

```

    inc(cumle_sayisi);

```

```

isim_sayisi:=0; //cumledeki isim kac tane
fiil_sayisi:=0;
diger_sayisi:=0;
noktadan_onceki_k:=k;
end
else //nokta degil ise cumle devam ediyor
if (a[k]<>' ')then
begin
kelimeler[harf]:=a[k];
inc(harf);
if (k=boyut_a) then goto sart2;
end
else //bosluk ile karsişlasınca...
sart2:
begin
noktadan_onceki_k:=k;
if kelimeler<>' ' then
begin
with form1.query1 do
begin
ara:=trim(kelimeler);

DisableControls;close;params[0].AsString:=strlower(pchar(ara));
open;enablecontrols;
end;

b:=form1.query1Anlam.value;
bulunanin_tipi:=form1.query1Tur.value;

if b="" then b:=kelimeler; //bulamazsa aynen kaldı...

if bulunanin_tipi='i' then
begin
inc(isim_sayisi);
isimler[isim_sayisi]:=b;
end
else
if bulunanin_tipi='y' then
begin
inc(fiil_sayisi);
fiil[fiil_sayisi]:=b;
end
else
begin
inc(diger_sayisi);
diger[diger_sayisi]:=b;

```

Sesli okuma kısmı:

```

unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  ExtCtrls, DBCtrls, Grids, DBGrids, Db, DBTables,
  StdCtrls, Buttons, WaveMix;

type
  ses = class(TThread)
  private
    { Private declarations }
  protected
    procedure Execute; override;

```

```

end;

inc(kelime_sayisi);

//if b<>" then kelimeler:=b;
//if ilk_kelime then turkcesatir:=trim(kelimeler)
// else turkcesatir:=turkcesatir+' '+trim(kelimeler); //ing aynen
kalir}
ilk_kelime:=false;

end;
harf:=1;
kelimeler:= ' ';
end;
end; //with k:=1
end; //for i:=0

end;

procedure cevir.Execute;
var bitti:boolean;
begin
  bitti:=false;

  repeat
    translate;

    bitti:=true;
    form1.query1.Active:=false;
    form1.SpeedButton1.Enabled:=true;
    form1.SpeedButton8.Enabled:=false;
    form1.SpeedButton9.Enabled:=false;
    form1.SpeedButton10.Enabled:=false;
    form1.Label2.Caption:='Kelime Sayısı='
    '+IntToStr(kelime_Sayisi);
    form1.Label3.Caption:='Cümle Sayısı='
    '+IntToStr(cumle_Sayisi);
    form1.Label5.Caption:='Çeviri Süresi='
    '+TimeToStr(now()-ilkzaman);

    until Terminated or bitti;
  end;

end.

```

```

procedure sesver;
end;

implementation

{ ses }

uses unit1;

procedure ses.sesver;
var i,j,k,l:integer;a:string;
begin
  j:=form1.richedit2.Lines.Capacity;
  for i:=0 to j do begin
    a:=lowercase(form1.RichEdit2.Lines.Strings[i]);
    if length(a)<>0 then begin

```

```

for k:=0 to length(a) do begin
if a[k]=' ' then
begin
form1.DXWaveList1.Items[0].Play(true);
end else
for l:=1 to 28 do begin //28 harfimiz var
if a[k]=harfler[l] then
begin
form1.DXWaveList1.Items[l].Play(true); break;
end;
end; // k:
end;
end; // i:
end;

procedure ses.Execute;
var bitti:boolean;
begin

```

```

bitti:=false;

repeat

sesver;

bitti:=true;

form1.SpeedButton13.Enabled:=true;
form1.SpeedButton15.Enabled:=false;
form1.SpeedButton16.Enabled:=false;
form1.SpeedButton17.Enabled:=false;

until Terminated or bitti;
end;

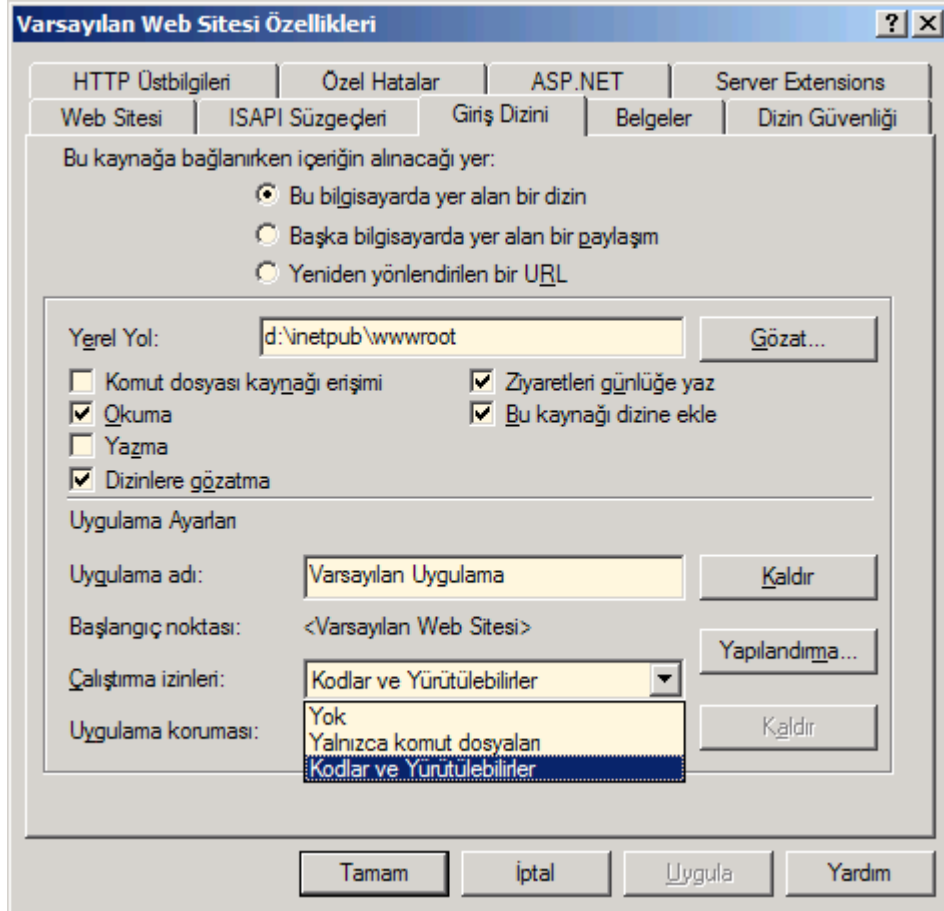
end.

```

27. CGI kullanımı(Common Gateway Interface)

(Ornekler\Delphi\delphiCGI.zip)

Byte Aralık 2004 sayısındaki bir örnek ile Delphi 7'nin 'Demos\Internet\CGIs' örneği birleştirerek güzel bir çalışma yapıyoruz. Sadece Perl veya C ile değil Pascal, VB ve Delphi ile de CGI uygulamaları rahatlıkla yapılabilir. Örneklerimizi IIS (Internet Information Service) programının kök dizini olan wwwroot içine açabilirsiniz. Aşağıdaki resimdeki gibi web sitesinde CGI çalışsın diye bir ayar da yapmamız(Çalıştırma izinleri) gerekli:



Resim 36a – "Varsayılan Web Sitesi"ne sağ tıklayın ve Özellikler deyin.

CGI uzantılı dosyaları yürütülebilir yani çalıştırılabilir dosya olarak görülmesi için bu ayarı yapıyoruz. Bu ayar dışına eğer CGI programımız dosya oluşturma ve değiştirme gibi işlemleri de yapıyor ise o klasörlere „Yazma“ hakkı da vermek gereklidir.

Kylix yani Delphi'nin Linux versiyonunda da aynı uygulama yapılabilir. Linux'da genellikle Apache sunucusu çalıştığı için, httpd.conf'da bazı ayarlar yaparak, programınızı aktifleştirebilirsiniz. Bu dosyadaki açıklamaları okumanızı tavsiye ederim.

CGI konsol uygulamalarınızı "Ornekler\Delphi\intrabob.zip" içindeki IntraBob.exe ile de test edebilirsiniz. Delphi'nin "Run*Parameters" ile Host Application belirtilir.

3 çeşit internet dosya tipi vardır: FTP (file transfer protocol) , HTML (Hyper Text Markup Language) ve CGI. CGI da standart konsol CGI ve Windows CGI olarak ikiye ayrılır. CGI ile sunucu üzerinde çalışan bir uygulamaya web tarayıcısı üzerinden veri iletişimi kurulur. Uygulamaya genellikle CGI script adı da verilir.

CGI uygulamalardaki sihirli kelime 'content-type: text/html' kısmıdır. Hemen ardından da boş bir satır eklenir. Böylece dosyamızın normal bir dosya değil, uygulama olduğunu sunucuya belirtiriz. Kod editörde yazıldıktan sonra, derlenerek sunucunun dizinine atılır. Dosya uzantısı ile de *.cgi olmasına gerek yok. *.exe olarak kalsa da çalışıyor.

CGI dosyamızı bir link(a href) ile, bir resim(img src) ile, bir form(action) ile veya doğrudan adresi(www) girerek çağırabiliriz.

En basit Delphi console CGI programı:

```
{$apptype console}
```

```
begin
  WriteLn('content-type: text/html');
  Writeln;
  WriteLn('Hello world!');
end.
```

Bir batch – yığın dosyası ile de CGI yapılabilir:

```
@ECHO OFF
ECHO content-type: text/html
ECHO.
ECHO ^<HTML^>^<HEAD^>^<TITLE^>^</TITLE^>^</HEAD^>^<BODY^>ECHO Hello world !
ECHO ^</BODY^>^</HTML^>
```

Delphi’de yeni bir „Console Application“ açıp şu kodları yazalım; Pascal üçgeninin kodları:

```
program byteCgi;

{$APPTYPE CONSOLE}

uses
  SysUtils;

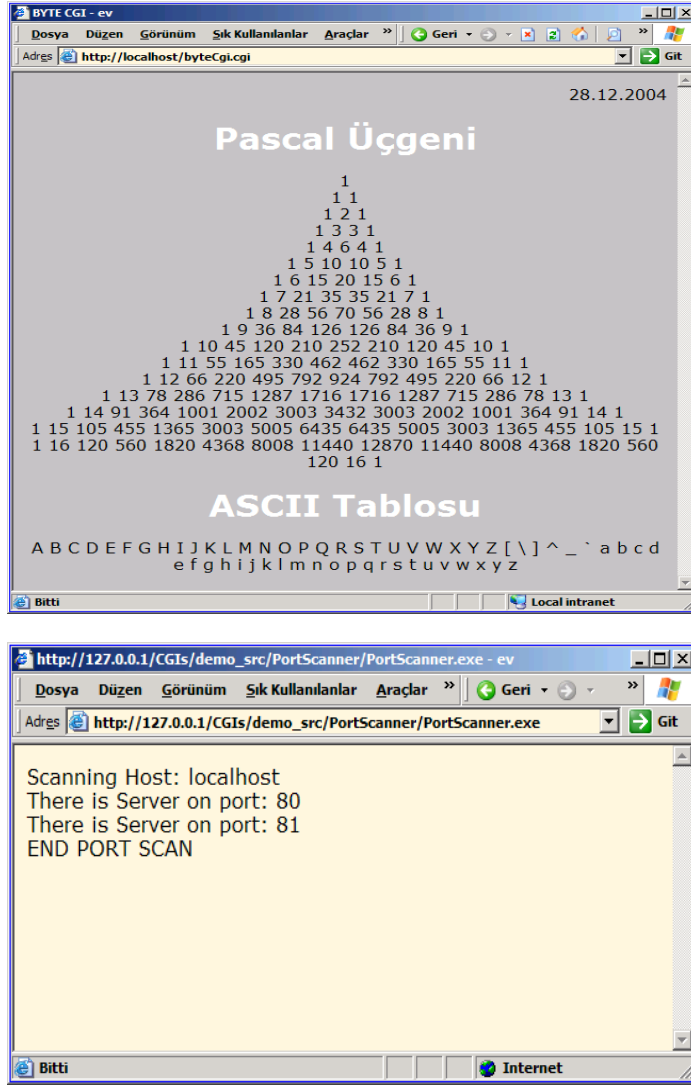
{$E cgi}

var
  pascal: array [1..17,1..17] of integer;
  satir,sutun,i,j: byte;
  karakter: integer;

begin
  writeln('content-type: text/html');
  writeln;
  writeln('<html>');
  writeln('<head>');
  writeln('<title>BYTE CGI</title>');
  writeln('</head>');
  writeln('<body bgcolor="#C0C0C0">');
  writeln('<p align="right">'+DateToStr(Now)+'</font></p>');
  writeln('<center>');
  writeln('<H1><font color="#FFFFFF">Pascal
  Üçgeni</font></H1>');
  writeln('<p>');
  pascal[1,1]:=1;

  for satir:= 2 to 17 do
  begin
    pascal[satir,1] := 1;
    for sutun := 2 to satir do
      pascal[satir,sutun] := pascal[satir-1,sutun] +
pascal[satir-1,sutun-1];
    end;
    for i:= 1 to 17 do
    begin
      for j:=1 to i do
      begin
        write(pascal[i,j]:7);
      end;
      writeln('<br>');
    end;
    writeln('</p>');

    writeln('<H1><font color="#FFFFFF">ASCII
    Tablosu</font></H1>');
    for karakter := 65 to 122 do
      write(chr(karakter)+' ');
    writeln;
    for i := 1 to ParamCount do
      writeln('<h1>' + ParamStr(i) + '</h1>');
    writeln('</center>');
    writeln('</body>');
    writeln('</html>');
  end.
```

Resim 36b – Basit örnek(Pascal Üçgeni) ve biraz daha ileri(Açık Port Tarama) bir uygulamamız...

Delphi 6 ve üstü sürümlerde Web Server Application olarak, hazır CGI uygulaması yapılabilir. WebModule içine Action'lar ekleyerek sayfaya parametreler yollanabilir. WebRequestCGI örneğinde çerez oluşturma ve okuma var. Daha gelişmiş uygulamaları File*New*Other...*IntraWeb*Standalone Application ile rahatlıkla yapılabiliriz. F9 ile sunucu programı çalıştırılır ve açılan pencereden de yine F9 ile sayfamızı çalıştırabiliriz. Boş form üzerinde Component Palette'den IW Standart, IW Data sayfalarından uygun nesneleri kullanabiliriz. Delphi 7'nin demolarını inceleyiniz.

28. Öğrenci Bilgi İşleri (paradox veritabanı örnekleri)

(Ornekler\Delphi\obi1.zip ve obi2.zip)

Okulda 2005 yılında bir döneme (3 ay) yayarak bu projeyi Delphi 6 ile geliştirdik. Not girişi ve devamsızlık girişi tam bitmedi. Alias yapmadığımızdan BDE kurulu bir bilgisayarda istediğiniz klasöre bu zip dosyayı açıp, çalıştırabilirsiniz. Kısaca şunları göreceksiniz: yazıcı kullanımı, sorgulama (SQL), form renkleri ve metin kutularının renk etkileşimi, açılış ekranı (splash form), ilk açılışta şifre sorarak giriş, kayıtlara resim ekleme... İkinci sürüm ise seminer çalışması olarak yaptık. OBI1'i sıfırdan tekrar başka yöntemler kullanarak oluşturduk. Her iki örnekte bitmeyen kısımlar var ise de, bana göre en büyük eksik, döküm kısmında 1. ve 2. dönem ortalamasının alınmamış olması. Karne dökümü en son iş olması gerekir. O da yok şimdilik. Mezunlar da bir tarafa tabi.

OBI1 daha çok görüntü özellikleri ile öne çıkıyor. Renkleri ve bazı ayarları aklında tutuyor. Yardım ve setup da olsaydı tam olacaktı. OBI2 ise kısa zamanda (6 gün – 18 saat) pratik ve minimalist yani az form ve az nesne ile çalışmanın ürünü.

Not hesabını bazen bildiğimiz FOR döngüsü ile, bazen tablolara calculated field – hesaplanan alan ekleyerek, bazen de SQL soruları ile ve son olarak da QReport içinde QRExpression nesneleri ile yaptık. Not çok önemli olduğu için, virgöl sonrası yuvarlamada çok dikkatli olmamız gerekiyor. 44.499 ile öğrencinin kalmaması gerekir değil mi? Ama bizde kalıyor. Hesaplama sırasında binde 5 gibi bir fazla sayı ile bu tür durumları engelleme de pek işe yaramaz. Güzel bir algoritma ile bu iş bitirilebilir.

Ama her zaman dediğim gibi güvenlik (şifreli giriş ve veritabanlarını koruma) ve çok kullanıcı çalışma eksiklerimiz var.

Veri Giriş

Ders Öğrenci Sınıf Okul

Arama

KayıtNo	Numarı	ÖğrenciAd	ÖğrenciSoyadı	ÖğrenciTelefon	ÖğrSınıf	ÖğrenciOğrenciMERNİS	ÖğrenciBabaM
1	1213	Ahmet	Çakar	()445-5555	K T11C	12345678901	Serbest Meslek
2	12	Mehmet	Keçi	(216)446-8877	K T10C	56564564586	Çalışmıyor
3	231	Ali	Veli	() -	E T12C	89898	
4	232	veli	sdf	() -	E T11C		
5	2323	ahmet	mehmet	()444-4444	K A9C		
6	2324	sasd	sadfsk	() -	A9C		

Öğrenci Adres

asd
asd
asd
asd

Öğrenci Resim



Öğrenci Hakkında Düşüce

asdasd
asdasdf
a-sd

Sınıf

T11C

Tamam

Not Devamsızlık Giriş

Not Devamsızlık

Öğrenci No ile Arama

ÖğrenciNo
1213

ÖğrenciAdı
Ahmet

ÖğrenciSoyadı
Çakar

ÖğrenciSınıf
T11C

Tarih Seçimi

Ağustos 2005

Devamsızlık Nedeni
Sabah geç kaldı

Devamsızlık Değeri
0,33

Toplam Devamsızlık: 0

Bugün: 28.08.2005

Tüm Devamsızlıklar

DevamOğrNO	DevamTarih	Devamsızlık
1213	24.05.2005	Sabah geç kaldı
1213	24.05.2005	Tam gün var
1213	24.05.2005	Sabah geç kaldı

Tamam

Resim 37 – OBI1 ve OBI2 program görüntüleri

29. Bit Filtrele (8 bit bilgiden seçim yapma)

(Ornekler\Delphi\bitislemleri.zip)

Okulda öğrencilere yaptığım basit bir örnek.



Resim 38 – Bit Filtrele

Onluk sayı sisteminde girilen, en fazla 255 değeri alabilen bir sayıdaki, belli bitleri ayıklayıp, sonuç kısmına yine onluk ve onaltılık olarak sayı yazan program. 8'e kadar istenen bitleri onay kutuları ile seçip , değerlerini sağa doğru kaydırarak tek sayı haline getiriyoruz. Mesela $(195)_{10}$ yani $(C3)_{16}$ sayısının ilk ve son bitini seçersek, sonuç 3 olur. Temel işlem komutları AND (ve ile maskeleme), SHR (sağa kaydırma), OR (sonuç değerlerini biraraya getirme).

Çöp bilgilerin atılıp, gerekli bilgilerin sadece kullanılması programı da diyebiliriz.

```
function hex(d:integer):string;
//DelphiTürkKodBankasından...
var kalan, sayi: integer;
    strKalan,sonuc:String;
begin
    sayi:=d;
    repeat
        kalan:= sayi mod 16;
        str(kalan,strKalan);
        case kalan of
            10: strKalan:='A';
            11: strKalan:='B';
            12: strKalan:='C';
            13: strKalan:='D';
            14: strKalan:='E';
            15: strKalan:='F';
        end;
        sonuc:= strKalan + sonuc ;
        sayi:=sayi div 16;
    until sayi < 1;
    hex:=sonuc;
end;
```

```
procedure TSDIAppForm.Button1Click(Sender: TObject);
var
    a,b,toplam:integer;
    bitler: array [1..8] of integer;
    secili:integer;
begin
    //Bütün programı buraya yazdık
    Button2Click(self);
    a:=strtoint(edit2.text);
    //10 luk sayı sisteminde giriş
    label4.Caption:= hex (a) ;
    //girişi hex yapıyoruz
    bitler[1]:=a and 1;
```

```
//a'nın 1. bitini alıyoruz
    label5.Caption:=(inttostr(bitler[1]));
    //1. bit ekrana yazıyoruz
    bitler[2]:=a and 2;
    label6.Caption:=(inttostr(bitler[2]));
    if label6.Caption='2' then label6.Caption:='1';
    //2. biti 1-0 şeklinde göster
    bitler[3]:=a and 4;
    label7.Caption:=(inttostr(bitler[3]));
    if label7.Caption='4' then label7.Caption:='1';
    bitler[4]:=a and 8;
    label8.Caption:=(inttostr(bitler[4]));
    if label8.Caption='8' then label8.Caption:='1';
    bitler[5]:=a and 16;
    label9.Caption:=(inttostr(bitler[5]));
    if label9.Caption='16' then label9.Caption:='1';
    bitler[6]:=a and 32;
    label10.Caption:=(inttostr(bitler[6]));
    if label10.Caption='32' then label10.Caption:='1';
    bitler[7]:=a and 64;
    label11.Caption:=(inttostr(bitler[7]));
    if label11.Caption='64' then label11.Caption:='1';
    bitler[8]:=a and 128;
    //a'nın 8. bitini alıyoruz
    label12.Caption:=(inttostr(bitler[8]));
    //ekrana 8. bit yazılır
    if label12.Caption='128' then label12.Caption:='1';
    //8. biti 1-0 şeklinde göster

    secili:=0;
    //kaç tane seçili onay kutusu var?
    if CheckBox1.Checked then inc(secili);
    if CheckBox2.Checked then inc(secili);
    if CheckBox3.Checked then inc(secili);
    if CheckBox4.Checked then inc(secili);
    if CheckBox5.Checked then inc(secili);
```

```

if CheckBox6.Checked then inc(secili);
if CheckBox7.Checked then inc(secili);
if CheckBox8.Checked then inc(secili);

toplamlam:=0;
//sonucu önce 0'lıyalım
for b:=secili downto 1 do begin
//yönü önemli değil aslında, kaç tane onay kutusu varsa...
if (CheckBox8.Checked)and(CheckBox8.Enabled) then
begin
bitler[8]:=bitler[8] shr (8-b);
//eğer 8. seçili ise sağa doğru kaydır

toplamlam:=toplamlam+bitler[8];
//sonucumuza bu yeni değerini ekle
CheckBox8.Enabled:=false;
//bu onay kutusunun işi bitti

end else
if(CheckBox7.Checked)and(CheckBox7.Enabled)then
begin
bitler[7]:=bitler[7] shr (7-b);
toplamlam:=toplamlam+bitler[7];
CheckBox7.Enabled:=false;
end else
if(CheckBox6.Checked)and(CheckBox6.Enabled) then
begin
bitler[6]:=bitler[6] shr (6-b);
toplamlam:=toplamlam+bitler[6];
CheckBox6.Enabled:=false;
end else
if (CheckBox5.Checked)and(CheckBox5.Enabled) then
begin
bitler[5]:=bitler[5] shr (5-b);
toplamlam:=toplamlam+bitler[5];
CheckBox5.Enabled:=false;
end else
if (CheckBox4.Checked)and(CheckBox4.Enabled) then
begin
bitler[4]:=bitler[4] shr (4-b);

```

```

toplamlam:=toplamlam+bitler[4];
CheckBox4.Enabled:=false;
end else
if(CheckBox3.Checked)and(CheckBox3.Enabled)then
begin
bitler[3]:=bitler[3] shr (3-b);
toplamlam:=toplamlam+bitler[3];
CheckBox3.Enabled:=false;
end else
if (CheckBox2.Checked)and(CheckBox2.Enabled)then
begin
bitler[2]:=bitler[2] shr (2-b);
toplamlam:=toplamlam+bitler[2];
CheckBox2.Enabled:=false;
end else
if (CheckBox1.Checked)and(CheckBox1.Enabled)then
begin
bitler[1]:=bitler[1] shr (1-b);
toplamlam:=toplamlam+bitler[1];
CheckBox1.Enabled:=false;
end;
end;
//ekrana sonucu yazdık
label2.Caption:=inttostr(toplamlam)+' ,hex: '+hex(toplamlam);
end;

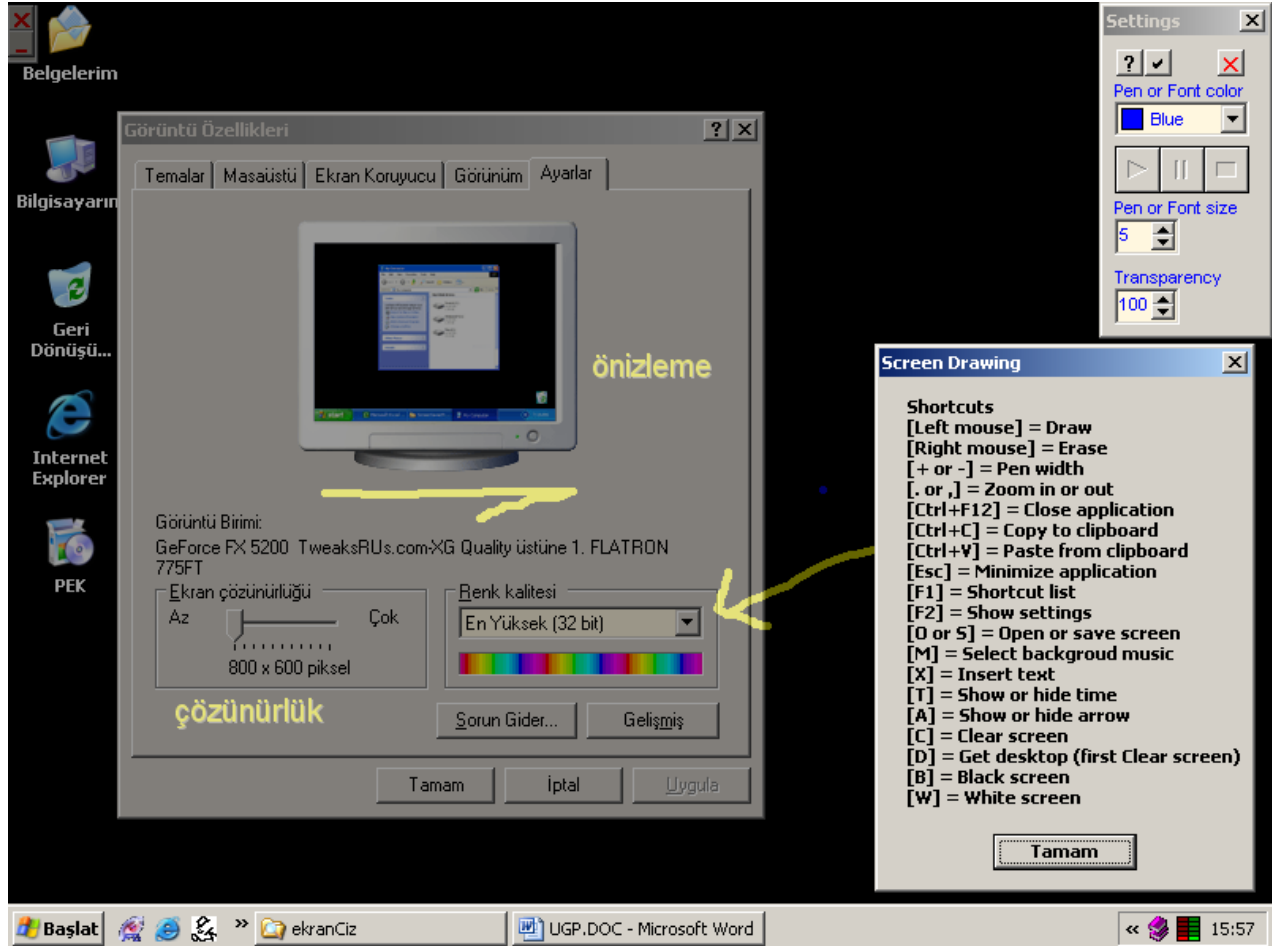
procedure TSDIAppForm.Button2Click(Sender: TObject);
//Yeni düğmesi
begin
//seçili uçları tekrar aktif ettik (reset)
CheckBox1.Enabled:=true;
CheckBox2.Enabled:=true;
CheckBox3.Enabled:=true;
CheckBox4.Enabled:=true;
CheckBox5.Enabled:=true;
CheckBox6.Enabled:=true;
CheckBox7.Enabled:=true;
CheckBox8.Enabled:=true;
end;

```

30. Ekran Çizimi (ekrana çizim yaparak, ders anlatımı)

(Ornekler\Delphi\ekranCiz2.zip)

Öğretmenlere yardımcı olmak için, ders anlatırken ekranda fare ile işaret etme yerine veya kalkıp lazer işaretleyicisi ile göstermek yerine, bu programı hazırladım. 500 satır gibi, fakat DelphiTurk KodBankası'ndan birçok hazır kod aldım ve geliştirdim.



Resim 39 – Ekran çiz

Ekran üzerine istenen renkte, istenen kalınlıkta kalem ile çizim yapmanın yanında, pano kullanımı, resim açma, kaydetme, boş ekran gösterme, saati gösterme, arkaplan müziği, fare oku gizleme gibi güzel özelliklerimiz var. Program menüsünde saatin yanına yerleşen simge ile de daha az yer kaplayan görünüm sağlanır. Hızlı geçişler için de kolay kısayollar ekledim. Özellikle Powerpoint'ten ESC, A, B, ve W kısayollarını örnek aldım.

Genel olarak iyi bir fikir, fakat tek kötü tarafı, XP ile desteklenmeye başlayan transparency (saydamlık - alphaBlend) özelliğinin biraz ekranı karartması. Eğer arkada çalışan programın çalışır halde gözükmesi önemli değilse, önce C ile ekranı temizleyip, D ile masaüstünün resmi getirilir. Böylece sanki tam ekran bir resim çizme programı gibi masaüstündeki programın görüntüsü üzerinde oynama yapabiliriz.

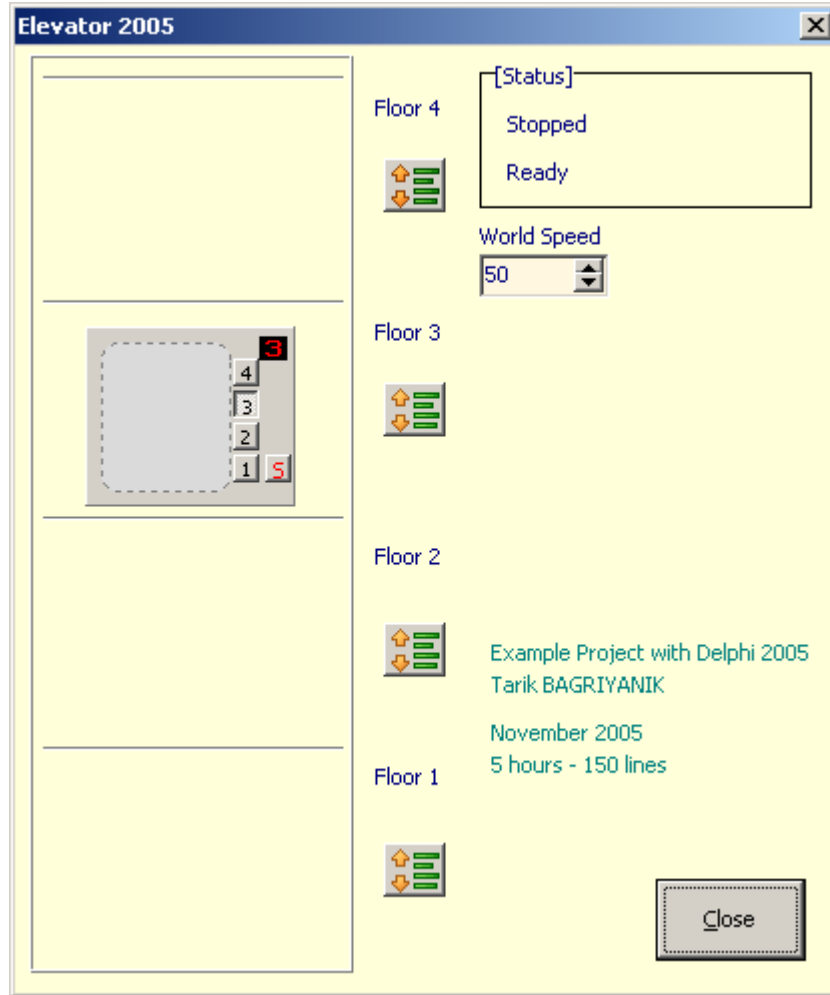
Yapılan hareketlerin geri alınabilmesi özelliği iyi olurdu. Ayrıca farklı fırça seçenekleri de yapılabilir. Dikdörtgen ve daire gibi yapılması fare ile zor olan şekilleri de kullanabilirdik. Başlat çubuğunu resim çizmeye dahil etmedik. Önceden hazırlanmış ders resimleri, bir klasöre kopyalanıp, istenirse slayt gösterisi gibi gösterilebilirdi. Boşluk ve BackSpace tuşları ile dersler arasında geçiş yapılabilirdi mesela. Türkçe dil desteği de lazım.

Projeksiyon ile ders anlatırken, makinenin Freeze (dondur) seçeneği olmalı. Bu arada siz de ekranı ayarlarsınız. Freeze'den çıkınca ekranda hazır bir şekilde yazılar ve resimler çıkar.

31. Asansör (4 kat binanın asansör simülasyonu)

(Ornekler\Delphi\asansor2.zip)

Okulda öğrencilerle yaptığım asansör örneği. 2 günde 5 saat uğraşıp, 150 satır kod yazdım. Thread (işlemci kanalları) ile yapılabilirdi. Aynı zip dosya içinde kanallı örnek de var.



Resim 40 – Asansör

Delphi 6-2005 arası rahatlıkla çalışır:

```
unit Unit3;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Variants, Classes,  
  Graphics, Controls, Forms,  
  Dialogs, ExtCtrls, Buttons, StdCtrls, Spin;
```

```
type
```

```
  TForm3 = class(TForm)  
    Timer1: TTimer;  
    Bevel1: TBevel;  
    Bevel2: TBevel;  
    Bevel3: TBevel;  
    Bevel4: TBevel;
```

```
    Bevel5: TBevel;  
    Panel1: TPanel;  
    Shape1: TShape;  
    SpeedButton1: TSpeedButton;  
    SpeedButton2: TSpeedButton;  
    SpeedButton3: TSpeedButton;  
    SpeedButton4: TSpeedButton;  
    StaticText1: TStaticText;  
    StaticText2: TStaticText;  
    StaticText3: TStaticText;  
    StaticText4: TStaticText;  
    Shape2: TShape;  
    SpeedButton5: TSpeedButton;  
    SpeedButton6: TSpeedButton;  
    SpeedButton7: TSpeedButton;  
    SpeedButton8: TSpeedButton;  
    SpeedButton9: TSpeedButton;
```

```

GroupBox1: TGroupBox;
StaticText7: TStaticText;
StaticText6: TStaticText;
StaticText8: TStaticText;
StaticText9: TStaticText;
StaticText10: TStaticText;
Button1: TButton;
SpinEdit1: TSpinEdit;
Label1: TLabel;
StaticText5: TStaticText;
StaticText11: TStaticText;
procedure SpinEdit1Change(Sender: TObject);
procedure SpeedButton4Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure Shape2MouseUp(Sender: TObject; Button:
TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Button1Click(Sender: TObject);
procedure SpeedButton9Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

```

```

var
  Form3: TForm3;
  counter : byte = 1;      //door counter
  requestFloor: byte=0;    //where to lift
  floor : byte = 1;        //lifting value
  state : byte = 1;        //1- stopped,
                           //2- door open(1 time),
                           //3- door wait(3 times),
                           //4- door close(1 time),
                           //5- move

```

implementation

```
{$R *.dfm}
```

```

procedure TForm3.FormCreate(Sender: TObject);
begin
  panel1.DoubleBuffered:=True; //smooth graphics
  state:=1;                    //stopped
  floor:=1;                    //active floor
  measure:=0;
end;

procedure TForm3.Timer1Timer(Sender: TObject);
begin
  //Elevator Engine

  if Form3.panel1.Top=360 then StaticText5.Caption:='1'
  else
  if Form3.panel1.Top=250 then StaticText5.Caption:='2'
  else
  if Form3.panel1.Top=140 then StaticText5.Caption:='3'
  else
  if Form3.panel1.Top=30 then StaticText5.Caption:='4';

  case state of
  1:
    begin

```

```

      counter:=0;
      StaticText6.Caption:='Stopped';
      StaticText7.Caption:='Ready';
    end;
  2:
    begin
      StaticText6.Caption:='Door Opening';
      StaticText7.Caption:='Not Ready';
      counter:= counter + 1 ;
      if counter = 12 then state:=3; //finished opening
      shape2.Width:=shape2.Width-5;
    end;
  3:
    begin
      StaticText6.Caption:='Door Opened';
      StaticText7.Caption:='Not Ready';
      counter:= counter + 1 ;
      if counter = 36 then          //12*3 more time waiting
        begin
          state:=4;                //close it
          counter:=0;
        end;
    end;
  4:
    begin
      StaticText6.Caption:='Door Closing';
      StaticText7.Caption:='Not Ready';
      counter:= counter + 1 ;
      if counter = 12 then state:=1; //finished closing
      shape2.Width:=shape2.Width+5;
    end;
  5:
    begin
      counter:=0;
      StaticText6.Caption:='Moving';
      StaticText7.Caption:='Not Ready';
      if (requestFloor=floor)or(measure=0) then
        begin
          state:=2;
          floor:=requestFloor; //new floor
          exit;                //cut the event!
        end; //same floor

      if ((requestFloor-floor)>0)and(measure<>0) then
        begin
          panel1.Top:=panel1.Top-5;
          measure:=measure-5;
        end;
      if ((requestFloor-floor)<0)and(measure<>0) then
        begin
          panel1.Top:=panel1.Top+5;
          measure:=measure-5;
        end;
    end;
  end;
end;

procedure TForm3.SpeedButton9Click(Sender: TObject);
begin
  //Emergency stop...
  timer1.Enabled:=not(timer1.Enabled);
end;

procedure TForm3.Button1Click(Sender: TObject);
begin
  close;                //end...
end;

```



```

procedure TForm3.Shape2MouseUp(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
  if state = 1 then state:=2; //door open if not moving...
end;

procedure TForm3.SpeedButton1Click(Sender: TObject);
begin //here we move!
  if state = 1 then begin
    requestFloor:=4; state:=5;
    measure:=(requestFloor-floor)*(panel1.height+20);
    if measure<0 then measure:=measure*-1;
    SpeedButton5.Down:=true;
  end;
end;

procedure TForm3.SpeedButton2Click(Sender: TObject);
begin
  if state = 1 then begin
    requestFloor:=3; state:=5;
    measure:=(requestFloor-floor)*(panel1.height+20);
    if measure<0 then measure:=measure*-1;
    SpeedButton6.Down:=true;
  end;
end;

```

```

procedure TForm3.SpeedButton3Click(Sender: TObject);
begin
  if state = 1 then begin
    requestFloor:=2; state:=5;
    measure:=(requestFloor-floor)*(panel1.height+20);
    if measure<0 then measure:=measure*-1;
    SpeedButton7.Down:=true;
  end;
end;

procedure TForm3.SpeedButton4Click(Sender: TObject);
begin
  if state = 1 then begin
    requestFloor:=1; state:=5;
    measure:=(requestFloor-floor)*(panel1.height+20);
    if measure<0 then measure:=measure*-1;
    SpeedButton8.Down:=true;
  end;
end;

procedure TForm3.SpinEdit1Change(Sender: TObject);
begin //timer speed
  timer1.Interval:=SpinEdit1.Value;
end;

end.

```

Bölüm 6 – Visual Basic

CD'deki "CD2\Programlama\VB\VB5CCEIN.EXE" yani Visual Basic Control Creation Edition ve Visual Basic 6.0 ile oluşturduğum örnekleri veriyorum. Bu bölüm için ipuçları ve benzeri bilgi yazamayacağım, Delphi kısmında bir kaç ipucu vermiştim. VB'cilere kolay gelsin, F5'e basın gitsin...

1. Sendkeys (diğer uygulamalara tuş basımı yollama)

(Ornekler\VB\sendkeys.zip)



Resim 1 – SendKeys formumuz

Private Sub Command1_Click()

```
npad = Shell("notepad.exe", vbNormalFocus)
AppActivate npad
SendKeys ("Hesap makinesinde bir hesap yapılıp, ")
SendKeys ("sonuç buraya kopyalanacaktır:" & Chr$(13)), True
```

```
clc = Shell("calc.exe", vbNormalFocus)
AppActivate clc
SendKeys Text1 & "-" & Text2 & "=", True
SendKeys "^C", True
If Check1 Then SendKeys "%{F4}", True
```

'True ile tuş yollama bitene kadar bekleme yapılır
'^ simgesi CTRL tuşudur
'% ile ALT tuşu taklit ediliyor

```
AppActivate npad
SendKeys ("Sonuç=" & "^v"), True
```

End Sub

Ne yaptık böylece? Not defteri ve hesap makinesi kendiliğinden açılıyor ve biz de istediğimiz tuş basımlarını bu uygulamalara yollayıp, işlem yapıyoruz.

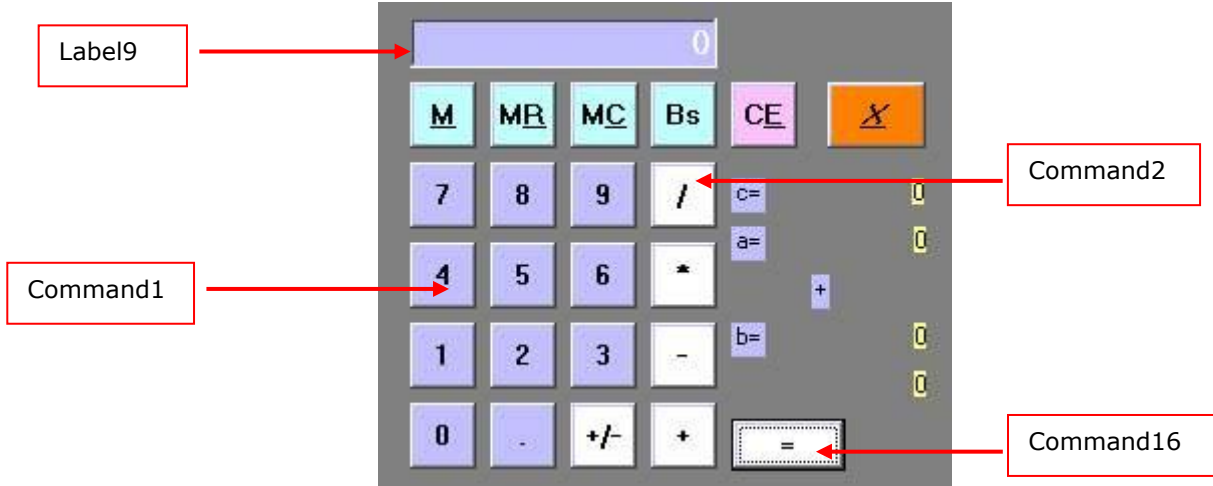
^ Control, % Alt, + Shift anlamına geliyor. {LEFT 42} deyimi 42 kere klavye sol tuşunun basımı demektir. PRTSC yani Print Screen tuş basımı programlara uygulanamaz. Diğer klavye tuşlarını VB veya Office VBA yardımıyla Sendkeys kısmında, Delphi CD'sindeki "Info\Extras\Sendkeys\sndkey32.pas" dosyasında da geniş açıklamalar ayrıca bulabilirsiniz. Dynamic Data Exchange (DDE) desteği veren Microsoft Windows programları dışında, bazı programlarda bu yöntem işe yaramayabilir.

Burada aklıma yeni bir fikir daha geliyor; paralel port, game port veya başka bir yoldan alabileceğimiz sinyalleri belli bir programa örneğin, bir oyuna tuş basımı yollamakta kullanabiliriz. Bir F-16 uçağının kokpit yapısı bir pano veya tabla üzerine simule – taklit edilerek, klavyenin yetersiz olduğu tuşların oyuna aktarılması yapılabilir. Asıl iş bir çok sayıdaki düğmeyi nasıl kodlayacağımız. Yine burada size bir açık kapı bırakıyorum.

2. Final (hesap makinesi yapımı)

(Ornekler\VB\Final\Project1.vbp)

Üniversitedeki finallere hazırlanırken yaptığımız hesap makinesi çalışmamız.



Resim 2 – Hesap Makinesi örneği

Hem fare ile hem de klavyeden tuş basımını kontrol ediyor. Bs Backspace tuşu, M hafıza - memory işlemleri, +/- ve "." kontrolü var. Form üzerindeki sadece CommandButton ve Label.

Option Explicit

Dim a, b As Double

Dim sonuc As Single

Dim hangi_sayi, noktali_a, noktali_b, arti_eksi_a, arti_eksi_b As Boolean

Dim islem As String * 1

Private Sub Command1_Click(Index As Integer) **'Numara ve nokta**

If Label9.Caption = "0" Then Label9.Caption = ""

If hangi_sayi = False Then

Label9.Caption = (Label9.Caption + Command1(Index).Caption)

Label1.Caption = Label9.Caption

If (Command1(Index).Caption = ".") And (noktali_a = True) Then

Label9.Caption = (Left\$(Label9.Caption, Len(Label9.Caption) - 1))

Label1.Caption = Label9.Caption

End If

If (Command1(Index).Caption = ".") And (noktali_a = False) Then noktali_a = True

Else

Label9.Caption = (Label9.Caption + Command1(Index).Caption)

Label2.Caption = Label9.Caption

If (Command1(Index).Caption = ".") And (noktali_b = True) Then

Label9.Caption = (Left\$(Label9.Caption, Len(Label9.Caption) - 1))

Label2.Caption = Label9.Caption

End If

If (Command1(Index).Caption = ".") And (noktali_b = False) Then noktali_b = True

End If

If Label9.Caption = "." Then Label9.Caption = "0."

Label3.Caption = Str(sonuc)

Label4.Caption = islem

Command16.SetFocus

End Sub

Private Sub Command1_KeyPress(Index As Integer, KeyAscii As Integer)

Command16.SetFocus

End Sub

```

Private Sub Command16_KeyPress(KeyAscii As Integer) '=
If Label9.Caption = "0" Then Label9.Caption = ""
If (KeyAscii = 42) Or (KeyAscii = 43) Or (KeyAscii = 45) Or (KeyAscii = 47) Then
    hangi_sayi = True
    islem = Chr$(KeyAscii)
    Label9.Caption = ""
    Label2.Caption = ""
    noktali_b = False
    Label3.Caption = "sonuc=" + Str(sonuc)
    Label4.Caption = islem
End If

If hangi_sayi = False Then
    If (KeyAscii > 47) And (KeyAscii < 58) Or (KeyAscii = 46) Or (KeyAscii = 8) Then
        If (Val(Label9.Caption) = 0) And (KeyAscii = 46) Then Label9.Caption = "0"
        If KeyAscii <> 8 Then
            Label9.Caption = (Label9.Caption + Chr$(KeyAscii))
            Label1.Caption = Label9.Caption
            If (KeyAscii = 46) And (noktali_a = True) Then
                Label9.Caption = (Left$(Label9.Caption, Len(Label9.Caption) - 1))
                Label1.Caption = Label9.Caption
            End If
            If (KeyAscii = 46) And (noktali_a = False) Then noktali_a = True
        Else
            If Len(Label9.Caption) <> 0 Then
                If Mid$(Label1.Caption, Len(Label1.Caption), 1) = "." Then noktali_a = False
                Label9.Caption = (Left$(Label9.Caption, Len(Label9.Caption) - 1))
                Label1.Caption = Label9.Caption
            End If
        End If
    End If
Else
    If (KeyAscii > 47) And (KeyAscii < 58) Or (KeyAscii = 46) Or (KeyAscii = 8) Then
        If (Val(Label9.Caption) = 0) And (KeyAscii = 46) Then Label9.Caption = "0"
        If KeyAscii <> 8 Then
            Label9.Caption = (Label9.Caption + Chr$(KeyAscii))
            Label2.Caption = Label9.Caption
            If (KeyAscii = 46) And (noktali_b = True) Then
                Label9.Caption = (Left$(Label9.Caption, Len(Label9.Caption) - 1))
                Label2.Caption = Label9.Caption
            End If
            If (KeyAscii = 46) And (noktali_b = False) Then noktali_b = True
        Else
            If Len(Label9.Caption) <> 0 Then
                If Mid$(Label2.Caption, Len(Label2.Caption), 1) = "." Then noktali_b = False
                Label9.Caption = (Left$(Label9.Caption, Len(Label9.Caption) - 1))
                Label2.Caption = Label9.Caption
            End If
        End If
    End If
End If
If Label9.Caption = "." Then Label9.Caption = "0."
End Sub

```

```

Private Sub Command19_Click() '+/-
If hangi_sayi = False Then
    If arti_eksi_a = True Then arti_eksi_a = False Else arti_eksi_a = True
    If (Val(Label9.Caption) <> 0) And ((Left$(Label9.Caption, 1) <> "-") Or (Left$(Label9.Caption, 1) <> "")) Then
        If arti_eksi_a = True Then
            Label9.Caption = ("-" + Label9.Caption)
            Label1.Caption = Label9.Caption
        Else
            If Left$(Label1.Caption, 1) = "-" Then
                Label9.Caption = (Right$(Label9.Caption, Len(Label9.Caption) - 1))
                Label1.Caption = Label9.Caption
            End If
        End If
    End If
End Sub

```

```

    End If
    End If
Else
    If arti_eksi_b = True Then arti_eksi_b = False Else arti_eksi_b = True
    If (Val(Label9.Caption) <> 0) And ((Left$(Label9.Caption, 1) <> "-") Or (Left$(Label9.Caption, 1) <> "")) Then
        If arti_eksi_b = True Then
            Label9.Caption = "-" + Label9.Caption
            Label2.Caption = Label9.Caption
        Else
            If Left$(Label2.Caption, 1) = "-" Then
                Label9.Caption = (Right$(Label9.Caption, Len(Label9.Caption) - 1))
                Label2.Caption = Label9.Caption
            End If
        End If
    End If
End If
End If
End If
End Sub
Command16.SetFocus
End Sub

```

```

Private Sub Command2_Click(Index As Integer) '+-*/
    hangi_sayi = True
    islem = Command2(Index).Caption
    Label9.Caption = ""
    Label2.Caption = ""
    noktali_b = False
    Label4.Caption = islem
    Command16.SetFocus
End Sub

```

```

Private Sub Command16_Click() '=
a = Val(Label1.Caption)
b = Val(Label2.Caption)
    If islem = "+" Then sonuc = a + b
    If islem = "-" Then sonuc = a - b
    If islem = "*" Then sonuc = a * b
    If (islem = "/" And (b <> 0)) Then sonuc = a / b
    Label9.Caption = Str$(CSng(sonuc))
Label1.Caption = (sonuc)
Label2.Caption = (b)
Label3.Caption = Str(sonuc)
Label4.Caption = islem
If hangi_sayi = True Then hangi_sayi = False
a = sonuc
If a < 0 Then arti_eksi_a = True
If InStr(Label9.Caption, ".") > 0 Then noktali_a = True
Command16.SetFocus
End Sub

```

```

Private Sub Command2_KeyPress(Index As Integer, KeyAscii As Integer) '+-*/
Command16.SetFocus
End Sub

```

```

Private Sub Command21_Click() 'CE
a = 0
b = 0
sonuc = 0
Label9.Caption = "0"
islem = "+"
Label4.Caption = islem
hangi_sayi = False
Label1.Caption = "0"
Label2.Caption = "0"
Label3.Caption = "0"
Label4.Caption = islem
noktali_a = False

```

```
noktali_b = False
arti_eksi_a = False
arti_eksi_b = False
Command16.SetFocus
End Sub
```

```
Private Sub Command21_KeyPress(KeyAscii As Integer) 'CE
Command16.SetFocus
End Sub
```

```
Private Sub Command23_Click() 'x
End
End Sub
```

```
Private Sub Command3_Click() 'BS
If hangi_sayi = False Then
    If Len(Label9.Caption) <> 0 Then
        If Mid$(Label1.Caption, Len(Label1.Caption), 1) = "." Then noktali_a = False
        Label9.Caption = (Left$(Label9.Caption, Len(Label9.Caption) - 1))
        Label1.Caption = Label9.Caption
    End If
Else
    If Len(Label9.Caption) <> 0 Then
        If Mid$(Label2.Caption, Len(Label2.Caption), 1) = "." Then noktali_b = False
        Label9.Caption = (Left$(Label9.Caption, Len(Label9.Caption) - 1))
        Label2.Caption = Label9.Caption
    End If
End If
Command16.SetFocus
End Sub
```

```
Private Sub Command4_Click() 'MR
If hangi_sayi = False Then
    Label1.Caption = Label8.Caption
    Label9.Caption = Label8.Caption
Else
    Label2.Caption = Label8.Caption
    Label9.Caption = Label8.Caption
End If
Command16.SetFocus
End Sub
```

```
Private Sub Command17_Click() 'M
Label8.Caption = Label9.Caption
Command16.SetFocus
End Sub
```

```
Private Sub Command18_Click() 'MC
Label8.Caption = "0"
Command16.SetFocus
End Sub
```

```
Private Sub Form_KeyPress(KeyAscii As Integer)
Command16.SetFocus
End Sub
```

```
Private Sub Form_Load()
a = 0
b = 0
islem = "+"
sonuc = 0
Label9.Caption = "0"
Label1.Caption = "0"
Label2.Caption = "0"
Label8.Caption = "0"
Label3.Caption = "0"
```

```

Label4.Caption = islem
hangi_sayi = False
noktali_a = False
noktali_b = False
arti_eksi_a = False
arti_eksi_b = False
End Sub

```

Basit bir hesap makinesi için ne çok kod var değil mi? Sadece fare ile düğmelerin tıklanması ile yapsak daha basit ve kısa olurdu. Fakat fare ile yapılanın klavye ile de yapılması demek işimizi ağırlaştırıyor. Biraz denerseniz hala hatalar bulabilirsiniz. Formun sağındaki etiket – label’larla da birinci ve ikinci girilen rakam ve hafızaya alınan rakam gösteriliyor.

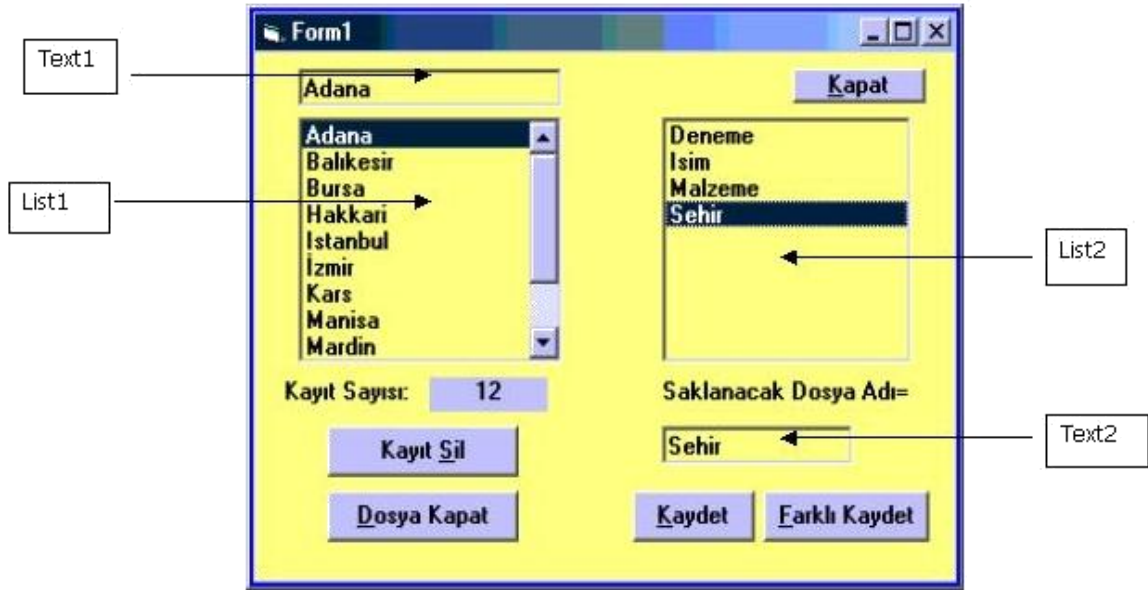
VB’de bir nesneyi kopyaladığınız zaman size “Zaten bu isme sahip nesne var. Dizi oluşturmak ister misiniz?” diye sorar. Biz de 0 ile 9 arasındaki ve “.” düğmelerini seri oluşturduk (Command1). +, -, * ve / düğmeleri de seri olacak (Command2). Bu sayede her düğmeye kod yazmak yerine tek kod ile yükümüzü hafiflettik. Hangi düğmeye tıklandığını Command1(Index).Caption gibi bir ifade ile anlıyoruz.

Command16.SetFocus ile de tüm hesaplamaları “=” tuşuna yönlendiriyoruz. Genel anlamda örneğin, **Command21_KeyPress(KeyAscii As Integer)**– tuş basımı, **Command18_Click()**– fare tıklaması olayları anlamına geliyor.

Aslında daha düzgün olması için form üzerine bıraktığınız nesnelere sayilar_Command, islem_Command, gosterge_Label gibi anlamlı isimler verebilirsiniz.

3. Listbox (Print ve Input komutları ile dosyalama)

(Ornekler\VB\Listbox\Project1.vbp)



Resim 3 – Listbox örneği

Dosyalama işlemlerine bir örnek. *.txt veritabanı oluşturmak, kayıt silmek, kayıt sayısı, kayıt güncelleme gibi işlemler yapılabilir. Çok ayrıntılı değil, zaten amaç basit dosyalama ve liste kutularının özelliklerini keşfetmek. Kayıt silme var ama siz de dosya silme eklemek istersiniz herhalde. Eğer List1 nesnesinin MultiSelect – çoklu seçimini, extended – gelişmiş yaparsanız kullanıcı birden fazla değeri Ctrl+fare sol tıklaması ile seçerek Kayıt Sil ile bunları silebilir. Kod olarak son seçilenden yukarı doğru silerseniz seçililerin indeks yani sıraları değişmemiş olur. Yani yukarıdan aşağıya silmeye çalışırsak satır numaraları devamlı değiştiği için istenen değerler silinemez.

VERSION 5.00

Begin VB.Form Form1

Appearance = 0 'Flat

BackColor = &H80000005&

Caption = "Form1"

ClientHeight = 3900

ClientLeft = 1395

ClientTop = 1380

ClientWidth = 5775

BeginProperty Font

Name = "MS Sans Serif"

Size = 8.25

Charset = 162

Weight = 700

Underline = 0 'False

Italic = 0 'False

Strikethrough = 0 'False

EndProperty

ForeColor = &H80000008&

LinkTopic = "Form1"

PaletteMode = 1 'UseZOrder

ScaleHeight = 3900

ScaleWidth = 5775

StartPosition = 2 'CenterScreen

Begin VB.CommandButton Command6

Caption = "&Dosya Kapat"

Height = 375

Left = 600

TabIndex = 9

Top = 3240

Width = 1575

End

Begin VB.CommandButton Command5

Caption = "&Kapat"

Height = 255

Left = 4440

TabIndex = 8

Top = 120

Width = 1095

End

Begin VB.TextBox Text2

Height = 285

Left = 3360

TabIndex = 6

Text = "DosyaAdi"

Top = 2760

Width = 1575

End

Begin VB.CommandButton Command4

Appearance = 0 'Flat

Caption = "&Farklı Kaydet"

Height = 375

Left = 4200

TabIndex = 5

ToolTipText = "Yeni isimle kayıt"

Top = 3240

Width = 1365

End


```

Begin VB.CommandButton Command3
    Caption      = "&Kaydet"
    Height       = 375
    Left         = 3120
    TabIndex     = 4
    ToolTipText  = "Sadece kayıt"
    Top          = 3240
    Width        = 1000
End
Begin VB.ListBox List2
    Height       = 1815
    Left         = 3360
    Sorted       = -1 'True
    TabIndex     = 3
    ToolTipText  = "Dosya listesi (Çift Tıklayın)"
    Top          = 480
    Width        = 2055
End
Begin VB.CommandButton Command1
    Caption      = "Kayıt &Sil"
    Height       = 375
    Left         = 600
    TabIndex     = 2
    Top          = 2760
    Width        = 1575
End
Begin VB.TextBox Text1
    Height       = 285
    Left         = 360
    TabIndex     = 1
    Top          = 120
    Width        = 2175
End
Begin VB.ListBox List1
    Height       = 1815
    Left         = 360
    MultiSelect  = 2 'Extended
    Sorted       = -1 'True
    TabIndex     = 0

```

```

    ToolTipText  = "Dosya içeriği"
    Top          = 480
    Width        = 2175
End
Begin VB.Label Label3
    BackStyle    = 0 'Transparent
    Caption      = "Kayıt Sayısı:"
    Height       = 255
    Left         = 240
    TabIndex     = 11
    Top          = 2400
    Width        = 1095
End
Begin VB.Label Label2
    Alignment    = 2 'Center
    Caption      = "Label2"
    Height       = 255
    Left         = 1440
    TabIndex     = 10
    Top          = 2400
    Width        = 975
End
Begin VB.Label Label1
    BackStyle    = 0 'Transparent
    Caption      = "Saklanacak Dosya Adı="
    Height       = 255
    Left         = 3360
    TabIndex     = 7
    Top          = 2400
    Width        = 2175
    WordWrap     = -1 'True
End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

```

Kod kısmı:

```

Dim ac As String
Private Sub Command1_Click() '---program buradan başlıyor
If List1.SelCount <> 0 Then
If List1.SelCount = 1 Then
    List1.RemoveItem List1.ListIndex
    Text1.Text = ""
Else ' işte geri geri silme kısmı
    For i = List1.ListCount - 1 To 0 Step -1
        If List1.Selected(i) = True Then
            List1.RemoveItem i
        End If
    Next i
    Text1.Text = ""
End If
End If
Label2.Caption = List1.ListCount
End Sub

Private Sub Command3_Click() 'SAKLA
dsy = ac & "\" & List2 & ".txt"
Open dsy For Output As #1
For i = 0 To List1.ListCount - 1
    List1.Selected(i) = True
    Print #1, List1.Text

```

```

    List1.Selected(i) = False
Next i
Close #1
Label2.Caption = List1.ListCount
End Sub

Private Sub Command4_Click() 'Farklı kaydet
Dim bulundu As Boolean
bulundu = False
dosyaadi = Text2.Text

For i = 0 To List2.ListCount - 1
    List2.ListIndex = i
    If List2.Text = dosyaadi Then
        MsgBox ("Dosya zaten var...")
        bulundu = True
        Exit For
    End If
Next i

If bulundu = False Then
    List2.AddItem dosyaadi

```

```

dsy = ac & "\dosyalar.txt"
Open dsy For Output As #1
For i = 0 To List2.ListCount - 1
    List2.ListIndex = i
    Print #1, List2.Text
Next i
Close #1

dsy = ac & "\" & dosyaadi & ".txt"
Open dsy For Output As #1
For i = 0 To List1.ListCount - 1
    List1.ListIndex = i
    Print #1, List1.Text
Next i
Close #1

End If
Label2.Caption = List1.ListCount
End Sub

Private Sub Command5_Click()
End
End Sub

Private Sub Command6_Click()
Text1.Text = ""
List1.Clear
Label2.Caption = List1.ListCount
End Sub

Private Sub Form_Load()
ac = App.Path
dsy = ac + "\dosyalar.txt"

```

```

Open dsy For Input As #1
Do While EOF(1) = 0
    Input #1, satir
    List2.AddItem satir
Loop
Close #1
Label2.Caption = List1.ListCount
End Sub

Private Sub List1_Click()
Text1.Text = List1.Text
End Sub

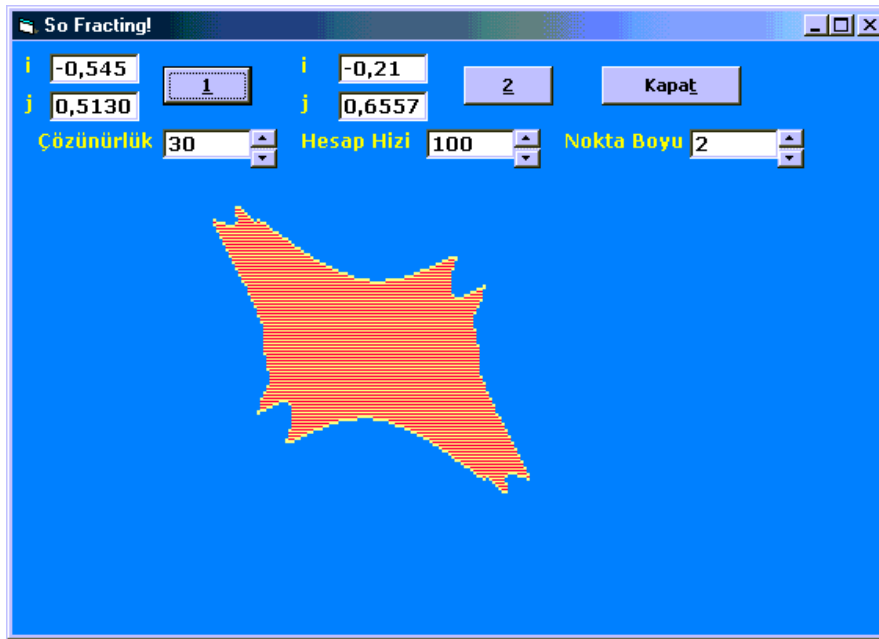
Private Sub List2_DblClick()
List1.Clear
Label2.Caption = List1.ListCount
dsy = ac & "\" & List2 & ".txt"
Open dsy For Input As #1
Do While EOF(1) = 0
    Input #1, satir
    List1.AddItem satir
Loop
Close #1
Text2.Text = List2
Label2.Caption = List1.ListCount
If List1.ListCount <> 0 Then List1.Selected(0) = True
Text1.Text = List1
End Sub

Private Sub Text1_KeyPress(keyascii As Integer)
If keyascii = 13 Then List1.AddItem Text1.Text
Label2.Caption = List1.ListCount
End Sub

```

4. Fractal (pset grafik komutu)

(Ornekler\VB\Fraktal\Project1.vbp)



Resim 4 – Karmaşık sayıların grafiği

VB dilinde de Pset (x, y) komutu ile form üzerine nokta basabiliriz. Bu örnekte kullanılan formül, karmaşık sayılar olarak bildiğimiz matematik konusuna dayanıyor. i ve j değerlerini kullanarak değişik fraktal görüntüler elde edebilirsiniz. Formun DrawMode özelliği kullanılarak noktaların stili belirlenebiliyor. Bölgesel Ayarlarda ondalık simgesi "." ise metin kutularını noktalı olarak değiştirmeniz gerekebilir. Şu anda "," virgül şeklinde. Çalışmıyor diye kızmayın!

VERSION 5.00

Begin VB.Form Form1

BackColor = &H00800000&

Caption = "So Fracting!"

ClientHeight = 5670

ClientLeft = 60

ClientTop = 285

ClientWidth = 8355

DrawMode = 8 'Xor Pen

DrawStyle = 1 'Dash

FillColor = &H00FFFF00&

FillStyle = 0 'Solid

BeginProperty Font

Name = "Tahoma"

Size = 9

Charset = 162

Weight = 400

Underline = 0 'False

Italic = 0 'False

Strikethrough = 0 'False

EndProperty

ForeColor = &H00FFFF00&

LinkTopic = "Form1"

ScaleHeight = 5670

ScaleWidth = 8355

StartPosition = 1 'CenterOwner

Height = 288

Left = 6480

TabIndex = 18

Text = "2"

Top = 840

Width = 852

End

Begin VB.VScrollBar VScroll3

Height = 375

Left = 7320

Max = 15

Min = 1

TabIndex = 17

TabStop = 0 'False

Top = 840

Value = 2

Width = 255

End

Begin VB.VScrollBar VScroll2

Height = 375

Left = 4800

Max = 500

Min = 1

SmallChange = 8

TabIndex = 12

TabStop = 0 'False

Top = 840

Value = 10

Width = 255

End

Begin VB.VScrollBar VScroll1

Height = 375

LargeChange = 10

Left = 2280

```

Max      = 500
Min      = 1
SmallChange = 8
TabIndex = 11
TabStop  = 0 'False
Top      = 840
Value    = 20
Width    = 255
End
Begin VB.TextBox Text6
    BackColor = &H80000009&
    Height    = 288
    Left      = 3960
    TabIndex  = 8
    Text      = "10"
    Top      = 840
    Width    = 852
End
Begin VB.TextBox Text5
    BackColor = &H80000009&
    Height    = 288
    Left      = 1440
    TabIndex  = 7
    Text      = "20"
    Top      = 840
    Width    = 852
End
Begin VB.TextBox Text4
    BackColor = &H80000009&
    Height    = 288
    Left      = 3120
    TabIndex  = 6
    Text      = "-0,11"
    Top      = 120
    Width    = 855
End
Begin VB.TextBox Text3
    BackColor = &H80000009&
    Height    = 288
    Left      = 3120
    TabIndex  = 5
    Text      = "0,6557"
    Top      = 480
    Width    = 855
End
Begin VB.CommandButton Command3
    BackColor = &H80000009&
    Caption    = "&2"
    Height    = 372
    Left      = 4320
    TabIndex  = 4
    Top      = 240
    Width    = 852
End
Begin VB.CommandButton Command2
    BackColor = &H80000009&
    Caption    = "Kapa&t"
    Height    = 375
    Left      = 5640
    TabIndex  = 3
    Top      = 240
    Width    = 1335
End
Begin VB.CommandButton Command1
    BackColor = &H80000009&
    Caption    = "&1"

```

```

Height    = 372
Left      = 1440
TabIndex  = 2
Top      = 240
Width    = 852
End
Begin VB.TextBox Text2
    ForeColor = &H00000000&
    Height    = 288
    Left      = 360
    TabIndex  = 1
    Text      = "0,11301"
    Top      = 480
    Width    = 855
End
Begin VB.TextBox Text1
    ForeColor = &H00000000&
    Height    = 288
    Left      = 360
    TabIndex  = 0
    Text      = "-0,74543"
    Top      = 120
    Width    = 855
End
Begin VB.Label Label7
    BackStyle = 0 'Transparent
    Caption    = "Nokta Boyu"
    ForeColor = &H0000FFFF&
    Height    = 375
    Left      = 5280
    TabIndex  = 19
    Top      = 840
    Width    = 1335
End
Begin VB.Label Label6
    BackStyle = 0 'Transparent
    Caption    = "j"
    ForeColor = &H0000FFFF&
    Height    = 255
    Left      = 2760
    TabIndex  = 16
    Top      = 480
    Width    = 135
End
Begin VB.Label Label5
    BackStyle = 0 'Transparent
    Caption    = "j"
    ForeColor = &H0000FFFF&
    Height    = 255
    Left      = 120
    TabIndex  = 15
    Top      = 480
    Width    = 135
End
Begin VB.Label Label4
    BackStyle = 0 'Transparent
    Caption    = "i"
    ForeColor = &H0000FFFF&
    Height    = 255
    Left      = 2760
    TabIndex  = 14
    Top      = 120
    Width    = 135
End
Begin VB.Label Label3
    BackStyle = 0 'Transparent

```

```

Caption      = "i"
ForeColor    = &H0000FFFF&
Height       = 255
Left         = 120
TabIndex     = 13
Top          = 120
Width        = 135
End
Begin VB.Label Label2
    BackStyle  = 0 'Transparent
    Caption    = "Hesap Hizi"
    ForeColor  = &H0000FFFF&
    Height     = 375
    Left       = 2760
    TabIndex   = 10
    Top        = 840
    Width      = 1095

```

Kod kısmı:

```

Option Explicit
Dim m, n As Integer
Dim i, x0, y0, x, y, z, x1, y1 As Double

Private Sub Command1_Click()
    c1 = Text1
    c2 = Text2
    car = Text5
    dig = Text6
    Form1.DrawWidth = Text7
    Form1.Refresh
    For m = 0 To 200
        x0 = -2 + m / 25
        For n = 0 To 100
            y0 = 2 - n / 25
            x = x0
            y = y0
            For i = 1 To dig
                x1 = x * x - y * y + c1
                y1 = 0.5 * x * y + c2
                x = x1
                y = y1
                z = x * x + y * y
                If z > 2 Then GoTo 10
            Next i
            PSet (Width / 2 + 700 - m * car, Height / 2 + 1400 - n * car)
10: Next n
        Next m
        Beep
    End Sub

Private Sub Command2_Click()
    End
End Sub

Private Sub Command3_Click()
    Form1.Refresh
    Form1.DrawWidth = Text7
    c1 = Text4
    c2 = Text3
    car = Text5
    dig = Text6
    For m = 0 To 200

```

```

End
Begin VB.Label Label1
    BackStyle  = 0 'Transparent
    Caption    = "Çözünürlük"
    ForeColor  = &H0000FFFF&
    Height     = 255
    Left       = 240
    TabIndex   = 9
    Top        = 840
    Width      = 1095
End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

```

```

x0 = -2 + m / 25
For n = 0 To 100
    y0 = 2 - n / 25
    x = x0
    y = y0
    For i = 1 To dig
        x1 = x * x - y * y + c1
        y1 = 0.5 * x * y + c2
        x = x1
        y = y1
        z = x * x + y * y
        If z > 2 Then GoTo 10
    Next i
    PSet (Width / 2 + 700 - m * car, Height / 2 + 1400 - n * car)
10: Next n
Next m
Beep
End Sub

Private Sub Text5_Change()
    VScroll1.Value = Text5
End Sub

Private Sub Text6_Change()
    VScroll2.Value = Text6
End Sub

Private Sub Text7_Change()
    VScroll3.Value = Text7
End Sub

Private Sub VScroll1_Change()
    Text5.Text = VScroll1.Value
End Sub

Private Sub VScroll2_Change()
    Text6.Text = VScroll2.Value
End Sub

Private Sub VScroll3_Change()
    Text7.Text = VScroll3.Value
End Sub

```

Bölüm 7 – Diğer Diller ve çalışmalarım

Delphi dili kullanarak yaptığım ActiveX örneği, Excel makroları, Windows Script Host, Okul Ders Programı, Yunus, KüPro, MyBoard ve JavaScript örnekleri oluşturacağız. Elinizdeki basit imkanlar ile (not defteri ve Internet Explorer gibi) bile programlama yapabileceğinizi göreceksiniz.

1. Delphi ActiveX (web sayfasında ActiveX)

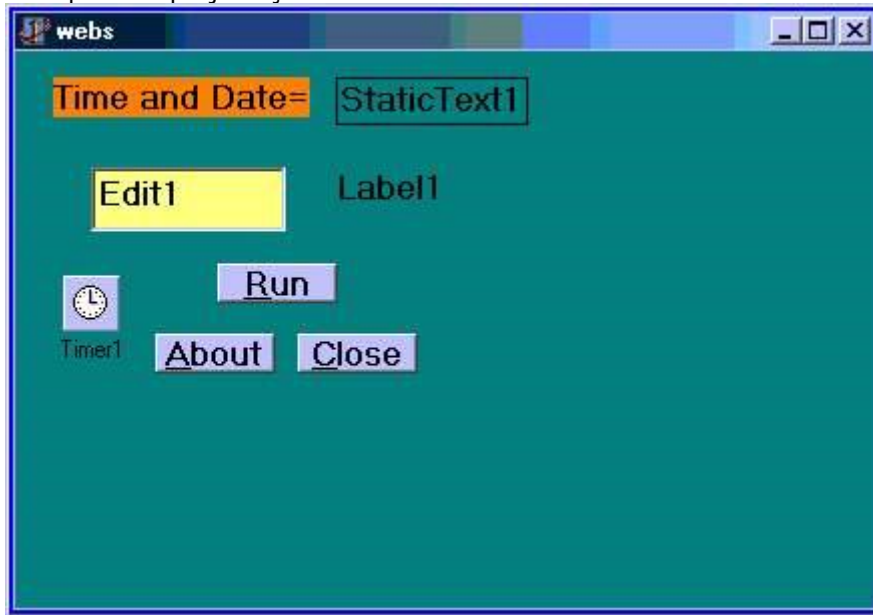
(Ornekler\Diger\ActiveX\websProj1.dpr)

ActiveX'i neden Delphi bölümüne koymadın dersiniz; bu dosyanın "websProj1.htm" dosyasını çalıştırdığınızda, "Windows\Downloaded Program Files" klasöründe "webs Control" adında bir dosya oluşturması. Bu oluşan "websproj1.ocx" artık hemen hemen tüm dillere gömülebilir hale geliyor. Asymetrics ToolBook Instructor adlı pek kullanmayı bilmediğim programlama diline, Visual Basic'e ve benzeri görsel dillere rahatlıkla "ActiveX Control" olarak dahil edilebiliyor. Yani sadece web sitesi içinde çalışmıyor, genel bir program haline geliyor. Netscape'de ActiveX'in çalışmadığını da unutmamak gerekir.

ActiveX denetimleri VB, Delphi ve VC denetimlerinden farklı değildir. Bir ActiveX denetimini onu kullanan uygulamanın içinde görebilir, onunla kapsayıcı uygulama penceresinde doğrudan iletişim kurabilirsiniz. Tipik Windows denetimleri mesaj tabanlı bir arabirim kullanır, VBX denetimleri özellik ve olayları kullanır, OLE Automation nesneleri özellik ve yöntemleri kullanır, ActiveX denetimleri de özellik, yöntem ve olayları kullanır. Bu üç eleman aynı zamanda Delphi bileşenlerinde de bulunur. ActiveX denetimleri ile doğrudan etkileşim kurulabilirken, OLE denetimlere çift tıklanarak (inside – out activation – içten dışa etkinleştirme) aktif edilerek kullanılır.

Delphi'de "View*Type Library" komutu ile oluşturduğumuz ActiveX denetiminin özellikleri görebiliriz. Aslında açılan bu pencerede websProj1.tlb dosyasını görüyoruz. *.tlb Type Library – tip kütüphanesi anlamına gelir.

Bizim yaptığımız Delphi'deki projede şunlar var:



Resim 1 – websProj formu

```
procedure Twebs.Button1Click(Sender: TObject);           // Run düğmesi
begin
label1.Caption:=edit1.text;
end;
```

```
procedure Twebs.Timer1Timer(Sender: TObject);           // Timer1'in işlevi
begin
StaticText1.Caption:=timetostr(time)+' '+DateToStr(date);
end;
```

```

procedure Twebs.Button2Click(Sender: TObject);           // A_bout düğmesi
begin About1.ShowwebsAbout; end;
procedure Twebs.Button3Click(Sender: TObject);           // C_löse düğmesi
begin close; end;

```

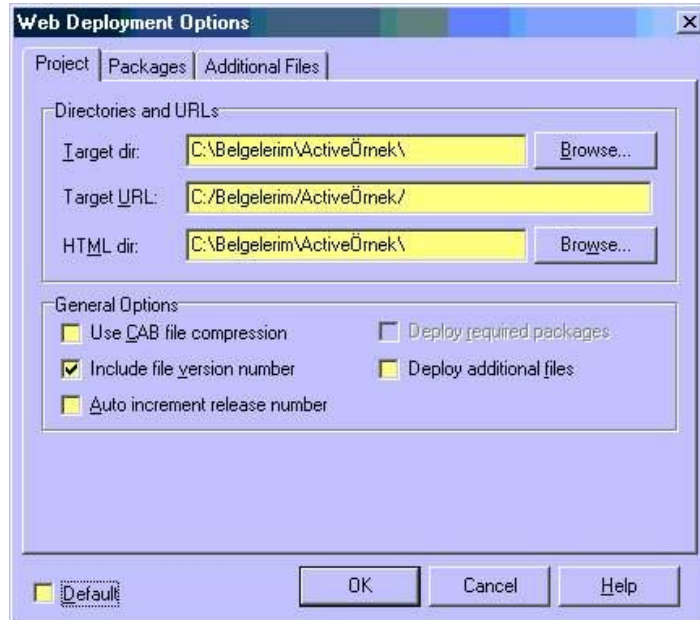
Delphi5'in Enterprise sürümünde (başka bir sürümde bu adımları farklı yöntemlerle yapabilirsiniz) ActiveX adım adım şöyle oluşturuluyor:

1. "File*New..." den ActiveX sekmesinden "ActiveForm" denilir.
2. Çıkan pencere şöyle doldurulur: ("Include About Box" ile hakkında kutusunu ekleyebilirsiniz)



Resim 2 – ActiveForm Sihirbazı

3. "Project*Web Deployment Options" doldurulur: ("Use CAB file compression" ile sıkıştırmayı deneyebilirsiniz)

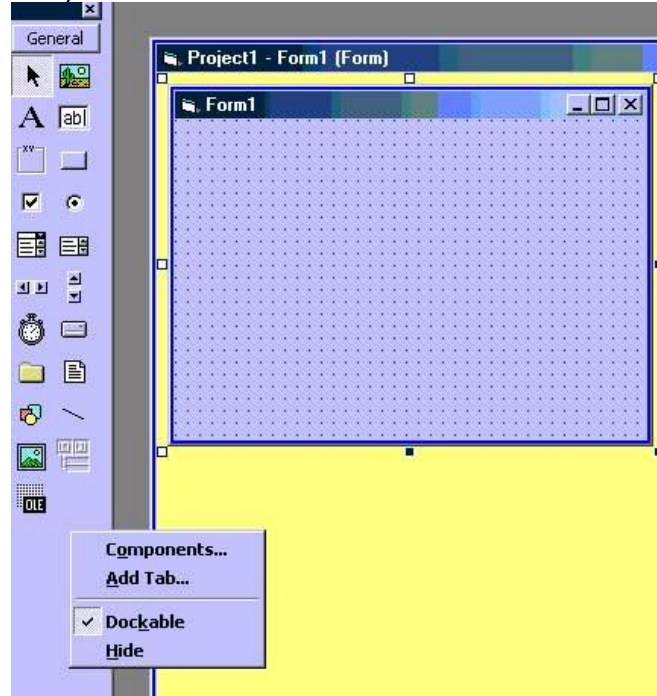


Resim 3 – Web yayımlama sihirbazı

4. Formun üzerine birkaç metin kutusu ve düğme atabilirsiniz.
5. "Run*Register ActiveX Server" komutu ile ActiveXDenemeProj1.ocx dosyamız kayıtlı hale gelir.
6. "Project*Web Deploy" komutu ile "c:\Belgelerim\ActiveÖrnek" klasörüne iki dosya oluşturulur. ActiveXDenemeProj1.htm ve ActiveXDenemeProj1.ocx.

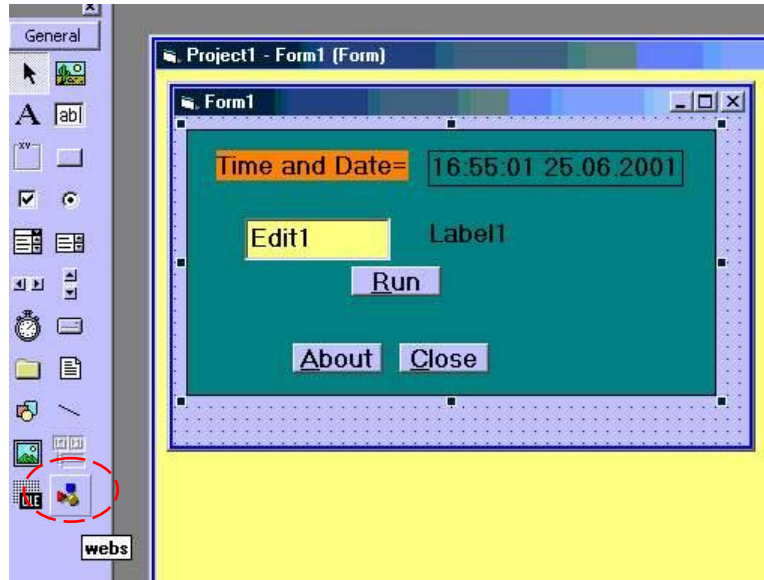
7. ActiveXDenemeProj1.htm'ye küçük bir müdahale etmemiz gerek. Dosyamızı Internet Explorer'da açın, boş bir yere sağ tıklayıp "Kaynağı Görüntüle" yapın. Aşağıdaki kısımdaki dizin ismini silerek klasör değişse bile ActiveXDenemeProj1.htm'sinin çalışmasını sağlayabilirsiniz. Eğer internette kullanmak isterseniz yani.
- codebase="ActiveXDenemeProj1.ocx#version=1,0,0,0"**

Visual Basic'i açıp şunu deneyelim:



Resim 4 – Visual Basic'de nesne ekleme

ToolBox'a sağ tıklayıp "Components..." komutunu verince listede websProj1 Library'sinin veya sizin deneme için yaptığınız ActiveXDenemeProj1 Library'sinin de bulunduğunu görürsünüz. Nesneleri ekleyin ve formun üzerine birini atalım. Sonuç:



Resim 5 – Sürpriz, Delphi VB içinde, ToolBox'taki nesnemize dikkat...

Tabii ki tam tersini de yapabilirsiniz. VB'de hazırlanmış bir *.ocx Delphi içine gömülebilir. İsterseniz Word veya Excel'de de VBA (Visual Basic for Applications)'i açarak makruza bu denetimi yani nesneyi ekleyebilirsiniz.

2. Excel yaziyla.xls (rakamı yazıya çevirme)

(Ornekler\Diger\Excel\yaziyla.xls)

	A	B	C	D	E
1	=yaziyla(5555)			beşbinbeşyüzellibeş	
2					
3					
4					

Resim 6 – Yazıyla makromuz

Bu Excel makrosu ile bir hücre içinde "=yaziyla(12345)" gibi komut vererek, "onikibinüçyüzkırkbeş" gibi yazı halini elde edebiliriz. Aynı mantıkla roma rakamları, İngilizce sayıları yazma yapılabilir. Makrolu bu dosyaları açarken çıkan uyarıya "Enable Macros" diyerek geçin.

"Tools*Macro*Visual Basic Editor" ile VBA (Visual Basic for Applications) açılır. Buradaki modül içindeki kodumuz:

```

'
' Yazıyla Macro
' Macro recorded 24.10.1997 by Tarık
'
Dim b$(9)
Dim y$(9)
Dim m$(4)
Dim v(15)
Dim c(3)

Function yaziyla$(sayi)
b$(0) = ""
b$(1) = "bir"
b$(2) = "iki"
b$(3) = "üç"
b$(4) = "dört"
b$(5) = "beş"
b$(6) = "altı"
b$(7) = "yedi"
b$(8) = "sekiz"
b$(9) = "dokuz"
y$(0) = ""
y$(1) = "on"
y$(2) = "yirmi"
y$(3) = "otuz"
y$(4) = "kırk"
y$(5) = "elli"
y$(6) = "atmış"
y$(7) = "yetmiş"
y$(8) = "seksen"
y$(9) = "doksan"
m$(0) = "trilyon"
m$(1) = "milyar"
m$(2) = "milyon"
m$(3) = "bin"
m$(4) = ""

a$ = Str(sayi)
If Left$(a$, 1) = " " Then poz = 1 Else poz = 0
a$ = Right$(a$, Len(a$) - 1)

For x = 1 To Len(a$)

```

```
If (Asc(Mid$(a$, x, 1)) > Asc("9")) Or (Asc(Mid$(a$, x, 1)) < Asc("0")) Then GoTo hata
Next x
```

```
If Len(a$) > 15 Then GoTo hata
```

```
a$ = String(15 - Len(a$), "0") + a$
```

```
For x = 1 To 15
```

```
v(x) = Val(Mid$(a$, x, 1))
```

```
Next x
```

```
s$ = ""
```

```
For x = 0 To 4
```

```
    c(1) = v((x * 3) + 1)
```

```
    c(2) = v((x * 3) + 2)
```

```
    c(3) = v((x * 3) + 3)
```

```
    If c(1) = 0 Then e$ = ""
```

```
    If c(1) = 1 Then e$ = "yüz"
```

```
    If c(1) > 1 Then e$ = b$(c(1)) + "yüz"
```

```
    e$ = e$ + y$(c(2)) + b$(c(3))
```

```
    If e$ <> "" Then e$ = e$ + m$(x)
```

```
    If (x = 3) And (e$ = "birbin") Then e$ = "bin"
```

```
    s$ = s$ + e$
```

```
Next x
```

```
If s$ = "" Then s$ = "sıfır"
```

```
If poz = 0 Then s$ = "eksi" + s$
```

```
yaziyla$ = s$
```

```
GoTo tamam
```

```
hata:
```

```
    yaziyla$ = "### hata"
```

```
tamam:
```

```
End Function
```

Kendi makronuzu yazmak isterseniz, boş bir Excel kitabında "Tools*Macro*Visual Basic Editor" ile VBA'yı açarak, önce "Insert*Module" komutu verin. Üstteki kodları buraya yazabilirsiniz. Sonra Excel'e geri dönerek yani VB'yi kapatarak, bir hücrede deneme yapabilirsiniz. Dikkat ederseniz **Function yaziyla\$(sayi)** şeklinde olan makromuz metin bir değer döndürüyor. "=yaziyla(2001)" yada bilgisayarın kafasını karıştırabilecek büyük rakamlar girin. Hataları görmeye çalışın. Virgüllü rakam yasak, negatif sayı da deneyebilirsiniz.

3. Excel dosyalistesi.xls (dosya arama)

(Ornekler\Diger\Excel\ dosyalistesi.xls)

	A	B	C	D
1				
2			e:\kitap\yardim içindeki tüm *.html listeleniyor...	
3			Klasör Adı	Dosya Adı
4			e:\kitap\yardim\	utility.html
5			e:\kitap\yardim\	programlama.html
6			e:\kitap\yardim\	diger.html
7			e:\kitap\yardim\	basic.html
8			e:\kitap\yardim\	c.html
9			e:\kitap\yardim\	delphi.html
10			e:\kitap\yardim\	assembler.html
11			e:\kitap\yardim\	pascal.html
12			e:\kitap\yardim\	visualbasic.html
13			e:\kitap\yardim\	digerdiller.html
14			e:\kitap\yardim\	kitapornek.html
15			Liste sonu ... 11 adet dosya var.	
16				

Resim 7 – Dosya listesini veren makromuz

Aşağıdaki örnekte de bir prosedür olarak belli bir klasör ve içindeki alt klasörlerindeki belli dosyaları (yani recursive – kendini çağıran, *rikörsiv diye okunur*) listeliyor:

Dim ilkhucure, sonhucure, dosyasayisi As Integer

```

Sub listele()
Dim dtipi$, klasor$
klasor = InputBox("Klasör Adı", "Neresi?", "c:\")
If klasor = "" Then End
dtipi = InputBox("Listelenecek dosya türü", "Tür?", "*.doc")
ActiveCell.Offset(0, 0) = klasor & " içindeki tüm " & dtipi & " listeleniyor..."
ActiveCell.Offset(1, 0) = "Klasör Adı"
ActiveCell.Offset(1, 1) = "Dosya Adı"
Call listele(klasor, dtipi, True)
ActiveCell.Offset(dosyasayisi + 2, 0) = "Liste sonu ... " & dosyasayisi & " adet dosya var."
End
End Sub

```

```

Sub listele(klasor$, dtipi$, alt%)
Dim klasorler(), i, dosya$, yol, attr%, ks%
Static r
On Error Resume Next
If Right$(klasor, 1) <> "\" Then klasor = klasor & "\"
If dtipi = "" Then End

```

```

dosya = Dir(klasor & dtipi, vbNormal)

```

```

Do While dosya <> ""
yol = klasor & dosya
ActiveCell.Offset(r + 2, 0) = klasor
ActiveCell.Offset(r + 2, 1) = dosya
dosyasayisi = dosyasayisi + 1
r = r + 1
dosya = Dir()
Loop

```

```

If alt = False Then Exit Sub
dosya = Dir(klasor & "*.*", vbDirectory)
Do While dosya <> ""
    attr = 0
    attr = GetAttr(klasor & dosya)

    If dosya <> "." And dosya <> ".." And (attr And vbDirectory) <> 0 Then
        ks = ks + 1
        ReDim Preserve klasorler(1 To ks)
        klasorler(ks) = dosya
    End If

    dosya = Dir()
Loop

For i = 1 To ks
    Call listeal(klasor & klasorler(i) & "\", dtipi, alt)
Next i
End Sub

```

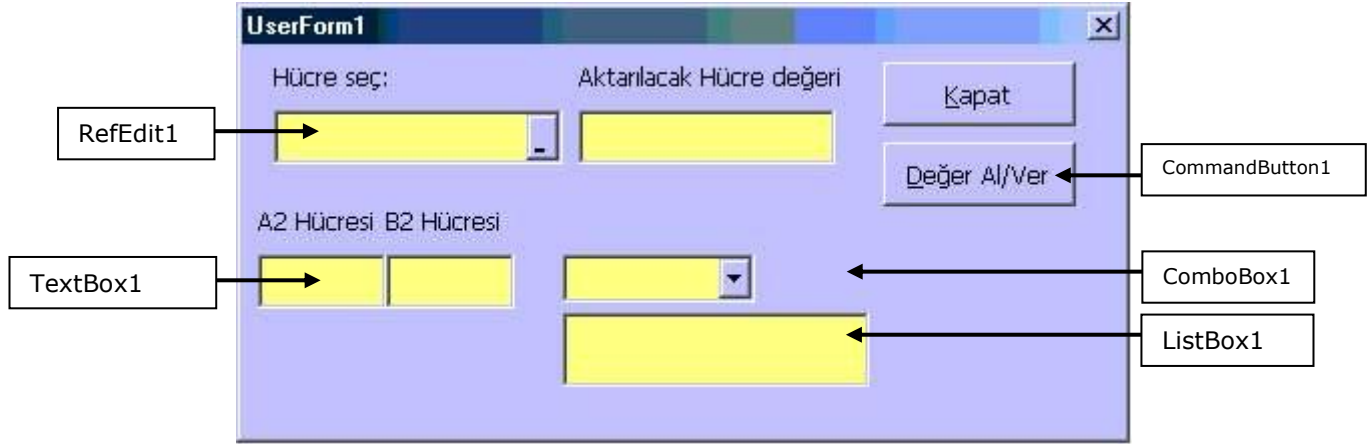
Farenin bulunduğu hücreden itibaren, bir klasördeki belli uzantıdaki dosya adlarını listeleyip, liste sonuna kaç dosya olduğunu yazan bir makro yaptık. Bu makro kodlarını alıp Delphi örneklerindeki Meraba1.0 ekran koruyucumuzda ve Gezgen örneklerinde kullanabiliriz. Delphi'deki Dosyalistesi örneğini de bu makroya bakarak hazırladım. Birkaç konu ilerde de, biraz abarttık ama, WSH ile liste.vbs adında bir örneğimiz daha var.

4. Excel userform.xls (kullanıcı tanımlı formlar)

(Ornekler\Diger\Excel\userform.xls)

Kendi formunuzu yapmak isterseniz Excel buna da izin veriyor. Yeni bir Excel kitabında "Tools*Macro*Visual Basic Editor" ile VBA'yı açarak, önce "Insert*UserForm" komutu verin. Aynen Visual Basic'deki gibi bir form olduğunu göreceksiniz. Office97'de VB 5.0, Office 2000'de VB 6.0, OfficeXP'de VB 6.3 kullanılarak makro yapılabilir.

Userform örneğimiz çok basit. Amaç forma hücre değerleri almak, formdan sayfamıza değer yollamak.



Resim 8 – Kullanıcı tanımlı form

ToolBox'a sağ tıklayıp "Additional Controls" diyerek "RefEdit.Ctrl" nesnesini ekleyin. Bununla hücre aralığı seçme nesnesini formumuza ekleyebileceğiz.

```
Private Sub ComboBox1_Click()
Sayfa1.Cells(2, 4) = ComboBox1.Text
End Sub
```

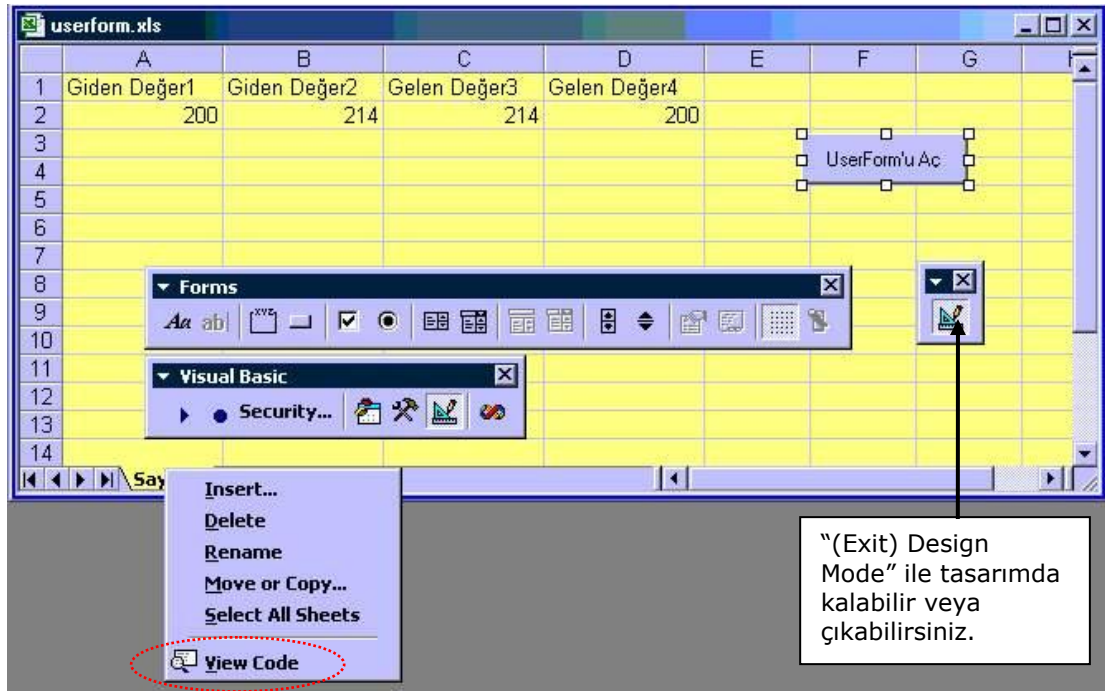
```
Private Sub CommandButton1_Click()
TextBox1.Text = Sayfa1.Cells(2, 1)
TextBox2.Text = Sayfa1.Cells(2, 2)
ComboBox1.Clear
ComboBox1.AddItem (Sayfa1.Cells(2, 1))
ComboBox1.AddItem (Sayfa1.Cells(2, 2))
ComboBox1.ListIndex = 0
ListBox1.Clear
ListBox1.AddItem (Sayfa1.Cells(2, 1))
ListBox1.AddItem (Sayfa1.Cells(2, 2))
If RefEdit1.Text <> "" Then Worksheets(1).Range(RefEdit1.Text).Value = TextBox3.Text
End Sub
```

```
Private Sub CommandButton2_Click()
Unload UserForm1
End Sub
```

```
Private Sub ListBox1_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Sayfa1.Cells(2, 3) = ListBox1.Text
End Sub
```

Bu arada Excel sayfamız üzerine bir düğme koyup bununla formumuzu açıyoruz. Bunun için Sayfa1, Sayfa2 diye sayfa isimlerinin bulunduğu sekmelere sağ tıklayıp "View Code" diyerek komutlarımızı yazabiliriz.

Bundan önce sayfa üzerine bir düğme atalım. Araç çubuklarına sağ tıklayıp "Forms" ve "Visual Basic" araç çubuklarının çıkmasını sağlayın. "Button" nesnesini atın, böylece CommandButton1'i oluşturdunuz.



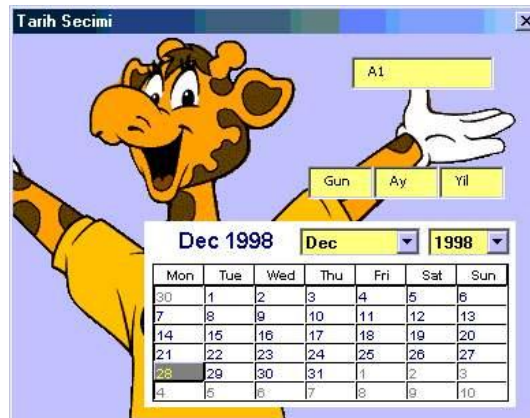
Resim 9 – Düğme ekleme

Oluşturduğumuz düğmeye "Design Mode" aktif iken çift tıklayıp şunu yazın:

```
Private Sub CommandButton1_Click()
UserForm1.Show
End Sub
```

Eğer böyle makro yapmak zor geliyor ise bir kısa yol daha var: Excel kendiliğinden zaten makro kaydediyor. "Tools*Macros*Record New Macro..." ile sık yaptığınız fare ve klavye hareketlerini tek bir komut haline getirip, araç çubuğuna bunu düğme olarak ekleyebilirsiniz. Tekrar "Tools*Macros*Macros...*Edit" diyerek VB düzenleyicisi ile müdahalelerde bulunabilirsiniz.

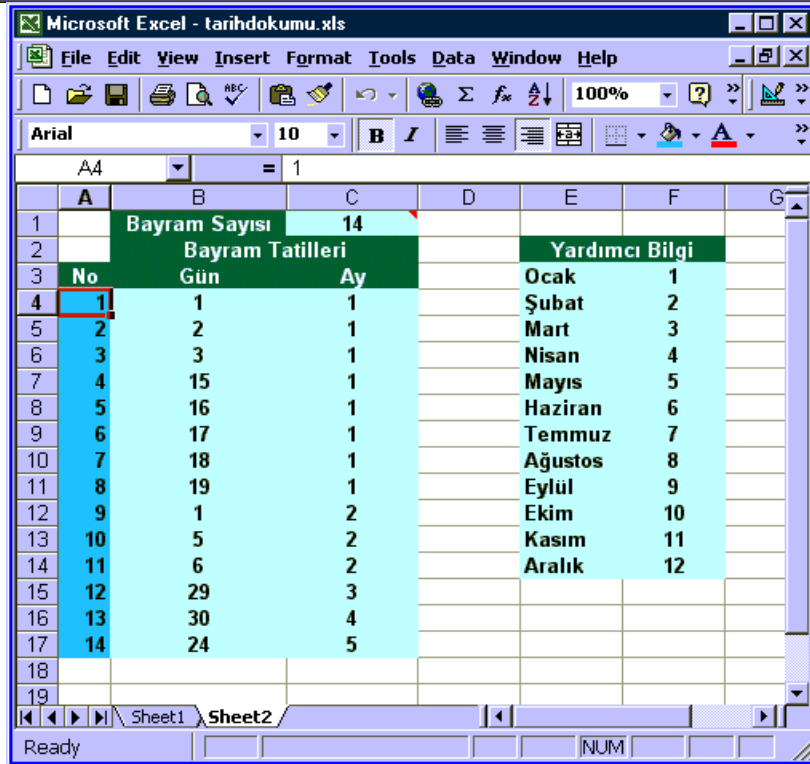
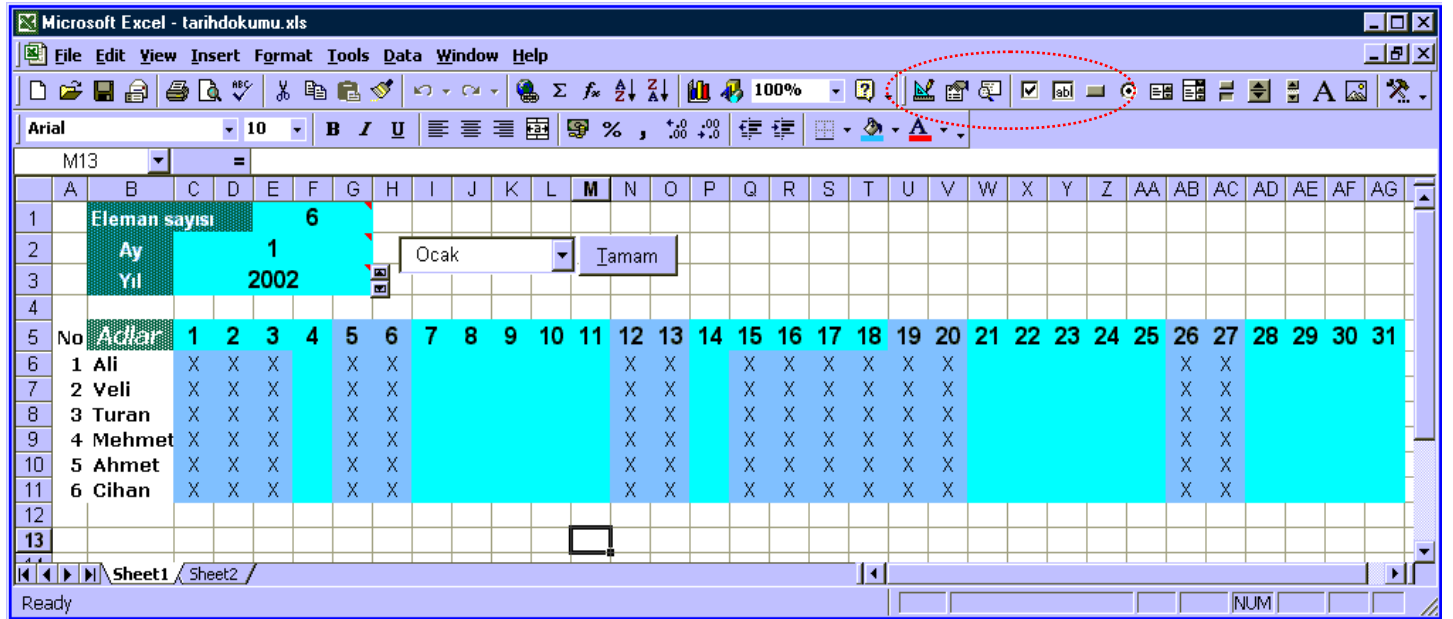
Office üyelerinin hepsi makro tabanlıdır, menü ve araç çubuğundaki komutlar aslında Excel veya Word'ün yazılım motoruna makro sinyallerini yollayan arabirimdir. Öyleyse VBA aslında bir eklenti değil Office paket programının ayakta durmasını sağlayan elemandır. Sık sık Excel kullanıyorsanız kendi hedefinize, zevkinize göre makrolar oluşturup işlerinizi hızlandırabilirsiniz. Tabii Excel, Word, PowerPoint ve Access'in makro komutları birbirinden farklıdır. Yani VBA Office üyeleri için ayrı ayrı davranır. Word makro kodlarını Excel'de kullanmak genellikle mümkün olmaz. tarih_göster.xls ve tarih_formu.xls de bu örneğe benzeyen UserForm'dan çalışma sayfasına tarih değeri aktaran makrodur. "A1" yazan yere "A1:C4", hatta "A1:C4, F3, G6" gibi koordinatlar belirtebilirsiniz. Calendar – takvim tıklanınca belirlenen aralıktaki hücelere tarih aktarılıyor.



Resim 10 – Kullanıcı tanımlı form – tarih_goster.xls

5. Excel ile tarih dökümü (modül kullanımı)

(Ornekler\Diger\Excel\tarihdokumu.xls)



Resim 11 – Excel’de tarih denetimi

Okulda öğretmenlerin ders saatlerinin tutulduğu çizelgede, belli bayram günleri ve tatil günlerini "X" ile dolduran bir programa ihtiyaç duyulmuştu. Bu program aynı zamanda takvim yapmak için de kullanılabilir. Kolay erişim için **gunliste()** makrosunu araç çubuğuna da ekleyebiliriz. Alt + F11 kısayolu ile Visual Basic Editor açık aşağıdaki kodları yazınız, daha sonra da araç çubuklarına sağ tıklayıp Customize... komutu ile Commands sekmesindeki "Macros*Custom Button" kısmından sürükleyip bırak ile düğme oluşturabiliriz. Yeni düğmeye sağ tıklayıp "Assign Macro..." ile gunliste() makrosunu atayabiliriz. İstediğiniz kadar eleman ve tatil günü listelenebilir. Kaç adet olacağı ile ilgili hücrelere değer verilerek yapılabilir. Ay için rakam girmek yerine ComboBox nesnesi de kullanılabilir.

```

Dim tarih As String
Dim tarih1 As String
Dim tarih2 As String
Dim son, i, j, k As Integer
Sub gunlistele()
    'resetleme ve hazırlık
    For i = 1 To 31
        Sheet1.Cells(5, i + 2).Interior.Color = RGB(0, 230, 255) 'iç rengi sıfırlanıyor
    Next i
    For i = 1 To 11
        For j = 6 To 11
            Sheet1.Cells(j, i + 2) = "" 'iki çift tırnak ile silme işlemi, burada belirli bir formül de olabilirdi
            Sheet1.Cells(j, i + 2).Interior.Color = RGB(0, 230, 200)
        Next j, i
        tarih1 = 1 & "/" & Sheet1.Cells(2, 3) & "/" & Sheet1.Cells(3, 3)
        tarih2 = 1 & "/" & (Sheet1.Cells(2, 3) + 1) & "/" & Sheet1.Cells(3, 3)
        son = DateDiff("d", tarih2, tarih1)
        If son < 0 Then son = son * -1
        If Sheet1.Cells(2, 3) = 12 Then son = 31
    'asıl işlemler buradan başlıyor
    For i = 1 To son
        tarih = i & "/" & Sheet1.Cells(2, 3) & "/" & Sheet1.Cells(3, 3)
        If Weekday(tarih, vbSunday) = 1 Or Weekday(tarih, vbSunday) = 7 Then
            Sheet1.Cells(5, i + 2).Interior.Color = RGB(100, 230, 255)
        End If
    Next i
    For i = 1 To son
        For j = 6 To 5 + Sheet1.Cells(1, 5) 'eleman sayısınca
            tarih = i & "/" & Sheet1.Cells(2, 3) & "/" & Sheet1.Cells(3, 3)
            If Weekday(tarih, vbSunday) = 1 Or Weekday(tarih, vbSunday) = 7 Then
                Sheet1.Cells(j, i + 2) = "X"
                Sheet1.Cells(j, i + 2).Interior.Color = RGB(100, 230, 255)
            End If
        Next j, i

        For k = 4 To 3 + Sheet2.Cells(1, 3) 'bayram sayısınca
            For i = 1 To son
                For j = 6 To 5 + Sheet1.Cells(1, 5) 'eleman sayısınca
                    tarih = i & "/" & Sheet1.Cells(2, 3) & "/" & Sheet1.Cells(3, 3)
                    tarih2 = Sheet2.Cells(k, 2) & "/" & Sheet2.Cells(k, 3) & "/" & Sheet1.Cells(3, 3)
                    If tarih2 = tarih Then
                        Sheet1.Cells(j, i + 2) = "X" 'eğer bayram ise de X konulur
                        Sheet1.Cells(j, i + 2).Interior.Color = RGB(100, 230, 255)
                    End If
                Next j, i, k
            End Sub
        End Sub

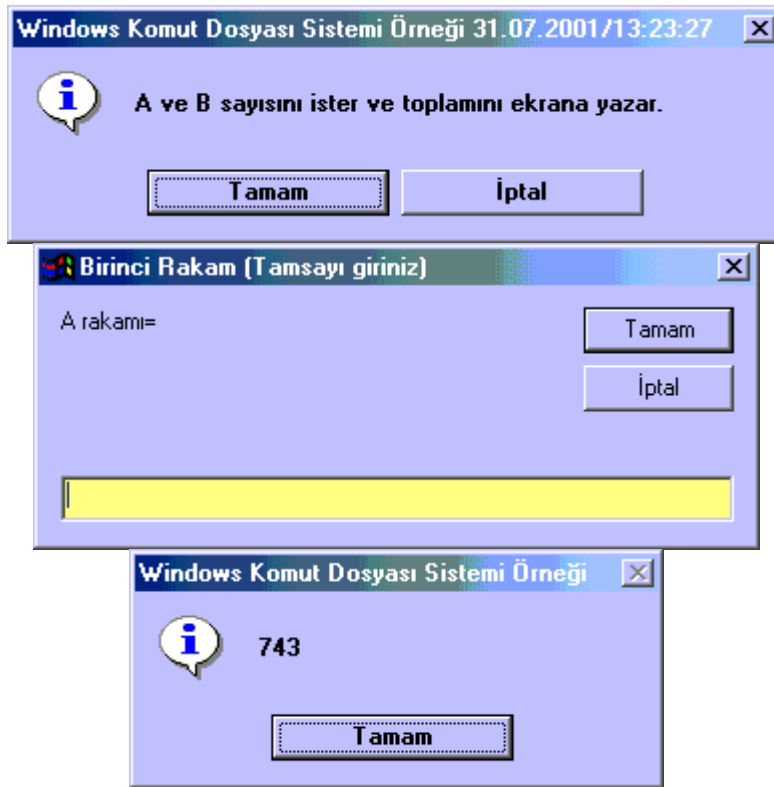
Private Sub CommandButton1_Click() 'Excel kitabı üzerindeki düğmenin kod kısmı
Gunlistele
End Sub

```

Excel dosyası üzerine görsel programlarda gördüğümüz düğme, çek kutusu, açılır kutu, liste gibi nesneleri yerleştirebilir, onları belli hücrelere bağlayabiliriz. Araç çubuklarına sağ tıklayıp "Control Toolbox" araç çubuğunu aktif ettiğinizde bu ve diğer eklenebilen nesnelere ulaşabilirsiniz. Örneğin düğme attığınızda bu nesneye sağ tıklayıp "Properties" diyerek özellikleri değiştirebiliriz.

6. WSH ile sayı toplama (*.vbs dosyalar)

(Ornekler\Diger\WSH\toplama.vbs)



Resim 12 – Windows Komut dosyası örneğimiz

Eğer Windows'unuzda WSH (Windows Scripting Host - Windows Komut Dosyası Sistemi) kurulu ise soyadı *.vbs (Visual Basic Script) veya *.js (J-Script) olan dosyaların üzerlerine çift tıklanıp çalıştırılabilir. J-Script Netscape firmasının geliştirdiği JavaScript'in benzeridir ve birbirleri ile uyumludurlar. Win98 için Program Ekle / Kaldır'dan WSH eklentisini yükleyebilirsiniz. Fakat WinME kendiliğinden yüklü olarak geliyor. Eğer başka bir program örneğin XingMPEG Player *.vbs dosyaları kendine yönlendirmiş ise o zaman ancak registry yedeğini geri alıp eski WSH bağına kazanabilirsiniz.

Bu arada script deyince aklıma Office içinde *VBA*, Corel içinde *CorelScript*, AutoCAD içinde *AutoLISP* ve Studio Max içinde de *MaxScript* gibi bu uygulamaların kendi yardımcı dilleri olduğu geliyor. Flash içinde de *Action Script* veya *Actions* – hareketler adlı C dili benzeri program yazma kısmı var. Herhangi biri ile uğraşıyor iseniz bu dilleri de öğrenmeniz yararınıza. Kendi komut yığınlarınızı oluşturup, işlevsel çalışabilirsiniz.

Bu örneği not defteri ile yazıp çift tırnak içinde "toplama.vbs" olarak kaydedin, sonra dosyanın üzerine çift tıklayın o kadar.

' Windows Komut Dosyası Sistemi Örnek Komut Dosyası

Mesaj = "A ve B sayısını ister ve toplamını ekrana yazar."

Baslık = "Windows Komut Dosyası Sistemi Örneği"

Call Toplama()

Dim WSHShell

Set WSHShell = WScript.CreateObject("WScript.Shell")

Sub Toplama()

Dim a

Dim b

Dim toplam
Dim yap

```
yap = MsgBox( Mesaj, vbOKCancel + vbInformation, Baslik & " " & date & "/" & time)
If yap = vbCancel Then WScript.Quit
```

```
a = inputbox( "A rakamı=", "Birinci Rakam (Tamsayı giriniz)" )
b = inputbox( "B rakamı=", "İkinci Rakam (Tamsayı giriniz)" )
sonuc = int(a) + int(b)
```

```
MsgBox sonuc, vbInformation, Baslik
End Sub
```

Microsoft'un asıl amacı bizi DOS içinde kullandığımız *.bat (batch – yığın dosyası) dosyalardan kurtarmak. Yani metin halindeki bu minicik dosyalar ile programlar açabilir, onlara özel komutlar yollayarak işlerimizi otomatikleştiririz. Tabii ki *.bat dosyadan farklı olarak çalışma anında size hatalı olan satırı rapor eden uyarıları da var. Çalıştırıldığında istenirse tüm işlemler artalandaki yapılabilir.

"Windows\Samples\WSH\chart.vbs" hoşunuza gidebilir. chart.vbs örneği WSH'nin Office ile olan entegrasyonuna bir örnektir. Bunun için COM - Component Object Model kullanıyor. COM ActiveX kontrolleri ve OLE için haberleşme temelini kurar. Böylece dev bir komut kümesi sizin emrinize sunulur. Windows veya DOS komutlarını da çalıştırabilirsiniz.

```
...
dim objXL
Set objXL = WScript.CreateObject("Excel.Application")
objXL.visible = true
objXL.Workbooks.Add
objXL.Cells(1,1).Value = 5
objXL.Cells(1,2).Value = 10
objXL.Cells(1,3).Value = "=a1+b1"
objXL.Cells(2,4).Formula = "=Max(a1:c1)"
objXL.Range("A1:C1").Select
objXL.selection.font.bold = true
objXL.Range("D2").Select
objXL.selection.font.italic = true
...
dim objWrd
set objWrd = Wscript.CreateObject ("Word.Application")
objWrd.visible = false 'ekrana Word gelmeyecek
objWrd.documents.add
objWrd.selection.TypeText ("Naber?")
objWrd.ActiveDocument.Saveas ("c:\windows\desktop\sil.doc")
objWrd.quit
...
set WshShell = Wscript.CreateObject("Wscript.Shell")
WshShell.Run ("command /c copy a:\*.* c:\temp") 'gizli dosyalar hariç kopyalayabilir
WshShell.Run ("notepad c:\autoexec.bat")
WshShell.Run ("command /c del c:\temp\*.*") 'merak etmeyin silmeden önce soru sorulacaktır...
...
dim objFile
set objFile = Wscript.CreateObject ("Scripting.FileSystemObject")
objFile.CopyFile "c:\*.*", "c:\temp", true 'gizli dosyalar dahil
...
```

Bu arada JavaScript için yazdığımız "JavaScript ile dosyalama" örneğinde aşağıdaki gibi satırlar var. Buradaki object ve scripting deyimleri üstte kullanılanlara çok benziyor.

```
var fso;
fso = new ActiveXObject("Scripting.FileSystemObject");
```

7. WSH ile dosya listeleme (*.vbs ile Excel otomasyonu)

(Ornekler\Diger\WSH\listele.vbs)

Windows Scripting Host kullanarak Excel içine belli uzantıdaki dosyalar listelenir. Maalesef *.* veya *.j* gibi joker karakterleri ile arama yerine üç harfli tam uzantı adı gerekiyor. Örneğin *.doc, *.bmp olabilir. Bu girişin son dört karakterine göre Excel sayfası içine Word dökümanlarının karakter sayısı ve boyutu, diğer dosyaların da sadece boyutu listeleniyor. Toplam kaç dosya olduğu ve kilobyte cinsinden toplam ekrana gelir. İsterseniz yanlarına oluşturulma tarihleri ve dosya nitelikleri gibi ayrıntıları da ekleyebilirsiniz. Listelemeyi ille de Excel'e yollamak yerine *.txt dosya haline getirip not defteri programına gönderilebilir. Hemen hemen her dilde dosya listelemeyi örneklemiş olduk. Windows'un Bul programı zaten yeterince hızlı ve ayrıntılı. Ama kendi programınız içinde buna benzer bir kodlamaya ihtiyaç duyabilirsiniz. *.vbs olan bu örneğimizi VB dilinde rahatlıkla kullanabilirsiniz.

```
dim objXL
dim objWrd
dim objFile
dim baslangic
dim i
dim karak
dim toplam
dim sayisi
```

```
baslangic = inputbox("Klasör
Adı","Klasör","c:\belgelerim")
uzanti = inputbox("Dosya uzantısı","Uzantı","*.doc")
if baslangic<>"" and uzanti<>"" then
on error resume next
'bir hata meydana gelirse ekrana mesaj çıkmaz
i = 2
function tekrar (ByVal p)
objXL.cells(1,5) = p
dim s
dim dizin
dim altdizin
dim dosya
set dizin = objFile.GetFolder(p)

for each dosya in dizin.Files
if ucase(right(dosya.name,4))=ucase(right(uzanti,4)) and
dosya.name<>"" then
objXL.cells(i,1) = dizin
objXL.cells(i,2) = dosya.name
objXL.cells(2,5) = dizin & "\" & dosya.name
if ucase(right(dosya.name,4))=".DOC" then
objWrd.documents.open(dizin & "\" & dosya.name)
karak = objWrd.ActiveDocument.characters.count
objWrd.ActiveDocument.close
objXL.cells(i,4) = karak
objXL.cells(i,3) = dosya.size\1024
'tamsayı sonuc elde etmek için \ kullanılabilir
toplam = toplam + dosya.size\1024
else
objXL.cells(i,3) = dosya.size\1024
toplam = toplam + dosya.size\1024
end if

i=i+1
```

```
sayisi = sayisi +1
end if
next

for each altdizin in dizin.subfolders
'i=i+1
s=s & tekrar (altdizin)
next

tekrar = s

end function

set objXL = Wscript.CreateObject ("Excel.application")
set objWrd = Wscript.CreateObject ("Word.application")
set objFile = Wscript.CreateObject
("Scripting.FileSystemObject")

objXL.workbooks.add
objXL.visible = true

objXL.cells(1,1).value = "Klasör"
objXL.cells(1,2).value = "Dosya Adı"
objXL.cells(1,3).value = "Dosya Boyutu (KB)"
objXL.cells(1,4).value = "Karakter Sayısı"
objXL.range ("a1:e1").select
objXL.selection.font.size = 16
objXL.selection.font.bold = true

objWrd.visible = false

tekrar (baslangic)

objXL.cells(1,5).value = "Arama Tamamlandı..."
objXL.cells(2,5).value = sayisi & " Adet Dosya " &
toplam & " KB"
objXL.cells.select
objXL.selection.columns.autofit
objXL.cells(1,1).select

objWrd.quit

else
msgbox ("Giriş boş olamaz...")
end if
```

8. kopyala komutu (batch ve Pascal dili kullanarak)

(Ornekler\Diger\kopyala.bat ve Ornekler\Pascal\okul\kopyala.pas)

*.bat dosyalardan biraz bahsedelim. DOS'da bir komut işlerken önce yerleşik – iç komut mu (dir, copy, del, ren gibi; format, sys, xcopy, ver gibi komutlar ise dış komut olarak adlandırılır), sonra bulunulan klasördeki *.com, *.exe veya *.bat dosya mı ve en son da path ile tanımlanmış klasörlerde mi diye aranır. Windows'un çalıştırılabilir dosyaları da *.com, *.exe, *.bat, *.cmd, *.vbs, *.vbe, *.js, *.jse, *.wsh, *.wsf'dir. Komut istemini açarak şunları deneyin. (↵ Enter demektir)

```
Kopyalar 2001 (c)
Kullanım "kopyala <dosyaadı> <sürücüadı>"
C:\TEMP>
Kopyalar 2001 (c)
İşlem = sil.gif dosyası, e: sürücüsüne kopyalanacaktır...
Emin misiniz? (Ctrl+C ile çıkış)
Devam etmek için herhangi bir tuşa basın . . .
^C

Toplu iş sonlandırılırsın mı (E/H)?e

C:\TEMP>kopyala sil.gif e_
```

Resim 13 – Kopyala.bat dosyamızın kullanımı

```
c:\> copy con kopyala.bat ↵
@echo off
cls
rem Bakalım nasıl olacak?
echo Kopyalar 2001 (c)
if "%1"==" " goto hata
if "%2"==" " goto hata
echo İşlem = %1 dosyası, %2: sürücüsüne kopyalanacaktır...
echo Emin misiniz? (Ctrl+C ile çıkış)
pause
if not exist %2:\%1 goto guvenli
echo %2:\%1 dosyası zaten var, üzerine yazılmaması için Ctrl+C yapın!
pause
:guvenli
copy %1 %2:\%1
echo "%1" kopyalandı!
goto son
:hata
echo Kullanım "kopyala <dosyaadı> <sürücüadı>"
:son
Ctrl+Z ↵

1 file(s) copied
c:\>kopyala deneme.doc d↵
Kopyalar 2001 (c)
İşlem = deneme.doc dosyası, d: sürücüsüne kopyalanacaktır...
Emin misiniz? (Ctrl+C ile çıkış)
Press any key to continue. . . ↵

d:\deneme.doc dosyası zaten var, üzerine yazılmaması için Ctrl+C yapın!
Press any key to continue. . . ↵

1 file(s) copied
"deneme.doc" kopyalandı!
c:\>
```

Eğer bu tip bir *.bat dosyayı kullanışlı bulup, herhangi bir yerde de bu komutu vermek isterseniz "kopyala.bat" dosyasını "Windows\Command" dizinine kopyalayın! Bunun sebebi sistemde tanımlanmış path – yollardan birinin de "Windows\Command" olması. Artık istediğiniz sürücü veya dizinde "kopyala *.* a" gibi bir komut

verebilirsiniz. Hatta “Başlat*Çalıştır...*kopyala yedek.* a” gibi de komut verebilirsiniz. Eliniz sürücü adından sonra “:” yazmaya alışmış ise birkaç satırı düzelterek bu desteği de verebilirsiniz.

Pascal dilinde de buna benzer bir program yapabiliriz:

```
program copyfile;
uses crt;
var sourcefile,destfile:file;
    recReads: word;
    buffer: array [1..1000] of byte;      {1000 byte'lik blok}
begin
    clrscr;
    if paramcount <> 2 then
    begin
        Writeln ('Kopyala 2001 (C)');
        Write ('Kullanım : kopyala kaynak_dosyaadi hedef_dosyaadi');
        halt;
    end;
    {$I-}
    assign (sourcefile, paramstr(1));
    reset (sourcefile,1);                {okumak için aç}
    if ioresult <> 0 then
    begin
        write (paramstr(1), ' bulunamadi...' ); halt;
    end;
    assign (destfile, paramstr(2));
    rewrite (destfile, 1);
    writeln ('Her # 1000 Byte anlamındadır...');
    blockread (sourcefile, buffer, sizeof(buffer), recReads);
    while recReads > 0 do
    begin
        write ('#');
        blockwrite (destfile, buffer, recReads);
        blockread (sourcefile, buffer, sizeof(buffer), recReads);
    end;
    close (sourcefile); close (destfile);
    writeln;writeln('Kopyalama tamamlandı.');
```

```
Kopyala 2001 (C)
Kullanım : kopyala kaynak_dosyaadi hedef_dosyaadi
C:\> kopyala c:\himem.sys d:\yedek.sys
Her # 1000 Byte anlamındadır...
#####
Kopyalama tamamlandı.
C:\>
```

Kaynak ve hedef dosya adı parametre olarak komut isteminde yazılıyor. Örnekteki gibi isim değiştirmek için kullanılan ren komutuna benzer de kullanabiliriz. “Ornekler\Pascal\Graf\plasma.pas” örneğinde de aynı dosyalama komutları, dışardan resim dosyası açmak amacı ile kullanılmış.

Pek MS-DOS’dan bahsetmek istemezdim ama birkaç ipucu verebilirim herhalde:

```
c:\>dir *.          : Sadece klasör adlarını listeler. (uzantısız dosyalar)
c:\>dir /ad         : Sadece klasör adlarını listeler.
c:\>dir /a-d        : Sadece dosya adlarını listeler.
c:\>subst z: c:\belgelerim: İstedığınız klasörü sürücü gibi yapabilirsiniz
c:\>copy con prn    : Ne yazarsanız printer – yazıcıya çıkarılır.
c:\>doskey dap = dir /on/p : dap isminde makro tanımlıyoruz.
c:\>dap ↵          : Dosya ve dizinler ad sırasında pause – aralık verilerek gösterilir.
                    doskey listesini görmek için F7 tuşuna basınız. Makrolarınızı görmek için “doskey /macros” deyin.
.      <DIR>         : İçinde bulunduğunuz dizin. Sadece simgesel gösterim.
..     <DIR>         : Bir üst dizin, sadece kök dizinde görüntülenmez.
```

Aslında hep yapmak istediğim ama bir türlü hala yapamadığım; birden fazla Windows kurulu olan makinede aralarında geçiş yapabilmek, yani boot menu oluşturmak. XP boot menüsünü diske otomatik olarak kurulurken bulaştırıyor. Eğer Win98 açılış disketi oluşturmuşsanız, bu disketteki config.sys'yi de inceleyebilirsiniz. Hatta bu diskette RAMDISK oluşturabilmek için gerekli komutarı da bulabilirsiniz. RAMDISK sayesinde RAM'inizin bir kısmını sürücü olarak atayabilirsiniz. Eğer en azından 512 MB RAM sahibiyseniz, swap dosyanızı RAM üzerine yönlendirebilmek PC'nizi oldukça hızlandırabilir.

config.sys içindekiler:

```
[MENU]
MENUITEM = Windows, Windows 98
MENUITEM = Windows_Old, Windows 98 Yedek
MENUcolor = 15,0
MENUdefault = Windows, 10

[Windows]
DEVICE=C:\WINDOWS\HIMEM.SYS
DEVICE=C:\WINDOWS\EMM386.EXE
device=C:\WINDOWS\COMMAND\display.sys con=(ega,,1)
Country=090,857,C:\WINDOWS\COMMAND\country.sys

[Windows_old]
DEVICE=C:\yedekWIN\HIMEM.SYS
DEVICE=C:\yedekWIN\EMM386.EXE
device=C:\yedekWIN\COMMAND\display.sys con=(ega,,1)
Country=090,857,C:\yedekWIN\COMMAND\country.sys
```

autoexec.bat içindekiler: (Millenium için ise c:\Windows\Command\cmdinit.bat içine yazılabilir)

```
@echo off
prompt $p$g
goto %config%
:Windows
SET windir=C:\WINDOWS
SET winbootdir=C:\WINDOWS
SET TEMP=C:\TEMP
goto end
:Windows_old
SET windir=C:\yedekWIN
SET winbootdir=C:\yedekWIN
SET TEMP=C:\TEMP
goto end
:end
```

msdos.sys içindeki bir iki satır ile manuel – el ile şu satırları güncelleyerek yapabiliyorum gibi. Ama açılışta menü çıkması diye bir şey yok tabii. DOS'dan açılış yapıp, Windows adı değiştirilerek diğer yedek Windows'un yeri belirtilebilir. Uzun ve sakıncalı bir yöntem, "Program Files" içindeki bazı programlar çalışmayabilir.

```
[Paths]
WinDir=C:\Windows          'burasına c:\yedekWin diyebilirsiniz...
WinBootDir=C:\Windows      'burasına c:\yedekWin diyebilirsiniz...
HostWinBootDrv=C
```

Eğer sık kullandığımız Control Panel – Denetim Masası öğelerine kısayol yapmak veya kendi programımızdan onları açmak istersek şu şekilde kullanabiliriz:

```
control appwiz.cpl          //program ekle / kaldır çalışır
control appwiz.cpl,, 2      //Windows kur menüsü
control appwiz.cpl,, 3      //başlangıç disketi oluşturma menüsü
control sysdm.cpl           //sistem bilgisi
control desk.cpl,,3         //Ekran ayarları açılır
control main.cpl,yazıcılar  //Yazıcılar klasörü açılır (yazıcılar yerine printers olabilir)
```

*.cpl dosyaları "Windows\System" dizininde bulabilirsiniz. Üzerlerine sağ tıklayıp "Denetim Masası ile aç" da diyebilirsiniz veya "Başlat*Çalıştır..." diyerek üstteki örnekleri deneyebilirsiniz.

Aşağıdaki gibi bir deneme.bat dosyası oluşturabiliriz. Buradaki /w ile işlem bitmeden diğerine geçilmemesi sağlanıyor:

```
start /w defrag /all
start /w scandisk /autofix
```

MS-DOS 'un temel eksikleri çok kullanıcı – multi user ve çok işli – multi tasking olmaması. Sadece PC – personal computer mantığı üzerine kurulu olan DOS, buna rağmen dünya üzerinde en çok kullanılan işletim sistemlerinden biri olmuştur. İlk kez 1968'de üretilen Unix ise bize internet ve hızlı bir ağ ortamı hediye etmiştir. Zaten çoğu www (http), ftp (file transfer protocol) ve SMTP (electronic mail) sunucusu Unix'dir. IBM OS/2 ve NT'den pek bahsetmek istemiyorum nedense. Microsoft Windows sayesinde, düşünsenize yüzlerce megahertz işlemcimiz, yüzlerce megabyte RAM ve gigabyte'larca sabit diskimizi yazı yazmak, oyun oynamak, internete bağlanmak için kullanıyoruz, hem de bu performansı maksimum kullanmadan. Zaten eminim meraklı kullanıcıların bilgisayarında muhakkak sadece Windows değil, Linux, BeOS gibi ikinci bir işletim sistemi de vardır! Marsa (kötü anlamda değil) gitmiş gibi hissetmek istiyorsanız, Microsoft işletim sistemleri dışındakileri de muhakkak deneyin.

Unix'de ise performans düşmeden bilgisayarın kaynakları birden fazla kullanıcı için paylaştırılabilir. Aynı bilgisayarda birçok terminal ekranı (virtual machine – sanal makine) birbirinden bağımsız olarak fonksiyon tuşları ile açılabilir. Ticari bir amaçla üretilmeyen, donanımdan bağımsız bu işletim sistemi üzerinde çalışmak üzere birçok program yazılmıştır. Unix Apple, Atari, Amiga gibi 68000 tabanlı, Sun Sparc, Alpha, MIPS, PowerPC ve tabii ki intel işlemcili bilgisayarlara kurulabilir. MS-DOS ise sadece intel (Cyrix ve AMD küçümsenmemeli) üzerine kurulabilir. Ancak NT ve Windows 2000'de birden fazla işlemci kullanılabilirken, Linux işletim sisteminin doğal olarak çoklu işlemci desteği vardır.

Uzun dosya isimleri de zaten Linux'da varken, Win95'den itibaren MS-DOS 8.3 formatını aşarak kullanıma açılmıştır. Dosya ve klasör erişim hakları FAT için imkansız iken, ancak NT ile NTFS – new technology file system ile bu imkan elde edilmiştir. Linux FAT üzerine de kurulabiliyor, kendi disk sistemi extended filesystem yani reiserfs, ext, ext2 ve ext3'dür. MS-DOS uygulamaları kontrolsüz olarak donanıma komut yollayabilirken, Linux'da yazılım ile donanım arasında kernel interface – çekirdek arabirimi bulunur. NT'de kullanıcı modu ve kernel modu ile daha çok kararlı sistem elde edilmiştir.

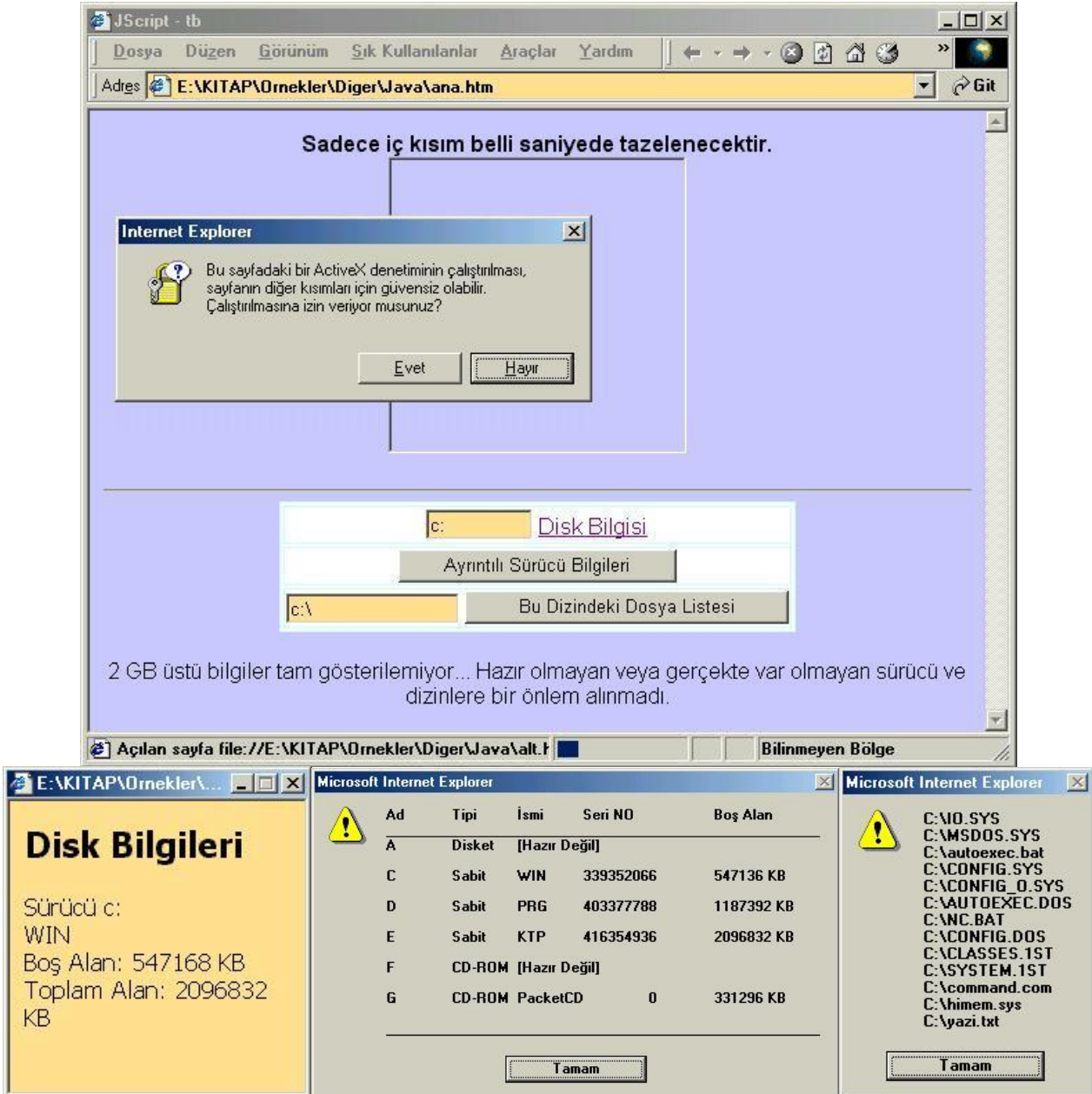
Swap yani geçici bellek alanı için Linux ayrı bir sürücü oluşturuyor. Oysa Windows boyu değişebilen win386.swp veya pagefile.sys dosyasını kullanıyor. Tabii ille de bu şekilde kullanmak zorunda değilsiniz. NT işletim sisteminde birden fazla disk daha performanslı veya kapasiteli kullanılması amacı ile, boş alanların birleştirilip sürücü oluşturulduğu volume set – hacim kümesi, her diskten aynı büyüklükte bölüm alınıp tek sürücü oluşturulduğu stripe set – şerit kümesi, birbiri ile eş sabit disklerin mirror set – ayna kümesi oluşturulması gibi RAID – Redundant Array of Independent / Inexpensive Disks (Fazladan bağımsız / ucuz disk dizisi) imkanları vardır.

MOD – Magneto Optik Disk bazı Machintosh sistemleri üzerinde standart olarak gelir. Bu disklerle de CD gibi optik, yani lazer ile veri yazma ve okuma, manyetik bir yüzey üzerinde gerçekleştirilir, fakat CD'den farklı olarak tekrar yazılabilirler. Yani sabit disk ve compact disk teknolojilerinin birleştirilmesi ile oluşan çıkartılabilir bir yedekleme ortamıdır. Flash RAM'ler ise genellikle fotoğraf makineleri gibi portatif aletlerin veri kaydı için kullanılır. Birçok firma birbiri ile uyumlu olmayan büyüklük ve kapasitede bu RAM'lerden üretilmişlerdir. Erişim süreleri nanosaniye seviyesindedir. Unix yeni çıkan teknolojilere uyum sağlayabiliyor. SCSI (*sıkazı diye okunur*), AGP ekran kartı veri yolu, USB seri port ve UDMA diskler gibi destekleri bulunuyor. Bu arada SCSI ile yazıcı, disk, tarayıcı gibi sekiz donanım parçası paralel olarak çalışabilir. Tek disketlik Unix'lerin yanında bir çok CD ve DVD'den oluşan Unix çeşidi vardır. Grafik arabirimi ilk Macintosh, sonra Windows ve X Window ile de Unix tanışmıştır. telnet (Telecommunications Network – Telekomünikasyon Ağı) ile başka bilgisayara bağlanarak bu bilgisayarı da kendi bilgisayarınız gibi kullanabilirsiniz. Çekirdeğinizi derleyerek daha hızlı çalışan Unix elde edebilirsiniz. İnternet sayesinde Unix donanım sürücülerinizi bularak onları güncelleyebilirsiniz. Modem, ses kartı, ekran kartı, ağ kartı, yazıcı gibi donanımlarınıza destek verip vermediğini internetten araştırabilirsiniz. Birden fazla ethernet – ağ kartı da destekliyor. Unix'de program nasıl kurulur, uninstall – silinir, defrag (var mı ki?), scandisk nasıl yapılır, donanım ayarları, kesmeler, donanım ekle nasıl yapılır, Türkçe karakter problemi nasıl çözülür, OpenGL (*oupin ci el diye okunur*) var mı, program nasıl yazılır, ne işime yarar bu Unix gibi soruları bana sormayın. CD'deki "Diger\Tutorial\Linux-HOWTOs.tar.gz" ile bu gibi sorularınıza cevap bulabilirsiniz. Sevdiğim halde ben de pek deneyimli değilim. Kusura bakmayın! Bu arada WinME gibi DOS komut isteminde açılmayan bir işletim sisteminiz ve C sürücüsünden farklı bir yerde olan Windows'unuz varsa kurulumda zorlanabilirsiniz. Yine de peşini bırakmayın. Pişman olmazsınız! Açılıştaki "LILO :'" diye kalmışsanız telaşlanmayın. Açılış disketi ile bir "fdisk /mbr" (mbr – Master Boot Record) yeterli. Fazla ayrıntıya girmeyeceğim, bunun yeri bu kitap değil.

9. JavaScript ile dosyalama (rasgele yazı kaydırılması)

(Ornekler\Diger\Java\ana.htm)

Hedefimiz bir *.txt dosyadan web sayfası içinde rasgele kayan yazılar oluşturmak. Ayrıca bilgisayarınızdaki disk ve dosya bilgileri de alınabiliyor. Sayfa ilk açılırken sorulan ActiveX güvenlik sorusuna Evet diyerek geçmemiz gerekiyor. Disk ile ilgili iki işlem yapıldığı için üst üste iki kere bu soru soruluyor. Biraz can sıkıyor gibi olsa da önemli bir soruşturmadır. İnternette dolaşırken böyle bir soruya Evet dersanız diskinize zarar veren komutlara da izin vermiş olursunuz. ActiveX komut ve nesneleri güvenlik yönünden çok zayıftırlar. Ayrıca Internet Explorer web tarayıcısı için güvenlik ayarlarından ActiveX ve JavaScript izinlerini bu örnekler için simdilik açmak gerekiyor. Deneyim için yaptığımız basit dosyalama için bu yöntemi tavsiye etmem, onun yerine cookie – çerez kullanımı daha rahat ve kolay.



Resim 14 – İlk açılış sorusu, ana.htm sayfası ve yan bilgiler

ana.htm içinde alt.htm dosya iframe (in line frame) – iç çerçeve (Netscape bu komutu desteklemiyor) içinde gösteriliyor. Bu sebeple iki *.htm dosya oluşturuyoruz. JavaScript C (i++; deyimi gibi), Pascal (var değişken tanımlama kısmı gibi) ile Basic (Array deyimi ile dizi tanımlamak gibi) dillerinin karışımı gibi gözüküyor. Rahatça değişken tanımlamak, istediğiniz yere kod yazmak, görsel diller kadar güçlü olan sayfa üzerindeki nesnelere

ulaşabilmek, bilgisayar kaynaklarını neredeyse tam kullanabilmek gibi birçok hoş özellik vardır. Eğer bilgisayarınızda Microsoft Visual Studio'nun bir ögesi kurulu ise web sayfası içinde yaptığınız hataları giderebilir, adım adım izleyebilir, komutlar hakkında bilgi alabilir, hem tasarım hem de ön izleme yapabilirsiniz. Internet Explorer içinden "Görünüm*Hata Ayıklayıcısı*Aç" komutu ile Microsoft Development Environment 6.0 programına ulaşabilirsiniz. Bu programı Office 2000 CD'sindeki "Add or Remove Features*Office Tools*HTML Source Editing" ile elde edebilirsiniz.

ana.htm ve alt.htm'ye ait kodlar:

```
<html>
<head>
<title>JScript</title>
<meta http-equiv="Content-Type" content="text/html; charset=">
</head>
<body bgcolor="#CCCCFF" OnUnload="kapat()">
<center>
<font face="Arial, Helvetica, sans-serif"><b>Sadece
iç kısım belli saniyede tazeleneyecektir.<br>
<iframe width="215" height="215" src="alt.htm" ></iframe>
</b> </font>
<form name="form1">
<hr>
<table border="1" align="center" bordercolor="#CCCCFF" bgcolor="#FFFFFF" bordercolorlight="#CCFFFF"
bordercolordark="#CCFFFF">
<tr>
<td>
<div align="center"><font face="Arial, Helvetica, sans-serif">
<input type="text" name="textfield" value="c:" maxlength="15" size="7" >
<a href="#" onMouseOver="bul()" onMouseOut="kapat()">
Disk Bilgisi</a></font></div>
</td>
</tr>
<tr>
<td>
<div align="center"><font face="Arial, Helvetica, sans-serif">
<input type="button" name="Submit2" value="Ayrıntılı Sürücü Bilgileri" onClick="listele()">
</font></div>
</td>
</tr>
<tr>
<td>
<div align="center"><font face="Arial, Helvetica, sans-serif">
<input type="text" name="textfield2" value="c:\\" size="15">
<input type="button" name="Submit3" value="Bu Dizindeki Dosya Listesi" onClick="dosyalar()">
</font></div>
</td>
</tr>
</table>
</form>
```

2 GB üstü bilgiler tam gösterilemiyor... Hazır olmayan veya gerçekte var olmayan sürücü ve dizinlere bir önlem alınmadı.


```
<script language="JavaScript1.2">
<!--Javascript kodunu eski sürüm browserlardan saklayalım
var fso;
fso = new ActiveXObject("Scripting.FileSystemObject");
Pencere = window.open("", "bilgiler", "top=0 left=0 width=200 height=200")
Pencere.close();
function bul()
{ var d, s;
Pencere = window.open("", "bilgiler", "top=0 left=0 width=200 height=200")
d = fso.GetDrive(fso.GetDriveName(document.form1.elements[0].value));
s = "Sürücü " + document.form1.elements[0].value + "<br>";
s += d.VolumeName + "<br>";
s += "Boş Alan: " + d.FreeSpace/1024 + " KB <br>" + "Toplam Alan: " + d.TotalSize/1024 + " KB";
```

```

        Pencere.document.write("<h2>Disk Bilgileri</h2>")
        Pencere.document.write("<p>" + s + "</p>")
        Pencere.document.close()
    }

function dosyalar()
{ var f, fc, s;
  f = fso.GetFolder(form1.elements[2].value);
  fc = new Enumerator(f.files);
  s = "";
  for (; !fc.atEnd(); fc.moveNext())
  {
    s += fc.item();
    s += "\n";
  }
  window.alert (s);
}

function listele()
{
  var s, n, e, x, t, d, kapasite , sn;
  e = new Enumerator(fso.Drives);
  s = "AdltTipiltismitSeri NOltBoş Alanln_____ln";
  for (; !e.atEnd(); e.moveNext())
  {
    x = e.item();
    switch (x.DriveType)
    {
      case 0: t = "Bilinmiyor"; break;
      case 1: t = "Disket"; break;
      case 2: t = "Sabit"; break;
      case 3: t = "Ağ"; break;
      case 4: t = "CD-ROM"; break;
      case 5: t = "RAM Diski"; break;
    }
    s = s + x.DriveLetter;
    s += " ";
    if (x.DriveType == 3)
      n = x.ShareName;
    else if (x.IsReady)
    { n = x.VolumeName;
      kapasite=x.FreeSpace/1024;
      sn = x.SerialNumber
      s += "\t" + t + "\t" + n + "\t" + sn + "\t"+ kapasite + " KB\n\n";
    }
    else
    {
      n = "[Hazır Değil]";
      kapasite=0;
      s += "\t" + t + "\t" + n + "\n\n";
    }
  }
  window.alert(s+"_____")
}

function kapat()
{
  Pencere.close();
}
-->
</script>
</font>
</center>
</body>
</html>

```

alt.htm'nin kodları. c:\yazi.txt veri dosyası yok ise oluşturuluyor.

```
<html><head>
<title>Yeni Belge</title>
<!--7 sn'de bir refresh için olabilirdi ... meta http-equiv="refresh" content="7"-->
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<meta http-equiv="Content-Language" content="tr">
</head>
<body bgcolor="#CCCCFF">

<SCRIPT LANGUAGE = "JavaScript1.2">
<!--Javascript kodunu eski sürüm browserlardan saklayalım
var arraym = new Array()
var dosya, ac, i, yazi;
var ForReading = 1; ForWriting = 2;

function dosyaAc()
{ dosya = new ActiveXObject("Scripting.FileSystemObject");
  if (dosya.FileExists("c:\\yazi.txt"))
  {
    ac = dosya.OpenTextFile("c:\\yazi.txt", ForReading);
    i = 0;
    while(!ac.AtEndOfStream)
    { arraym[i] = ac.ReadLine(); i++; }
    yaz();
  }
  else
  {
    ac = dosya.CreateTextFile("c:\\yazi.txt", ForWriting);
    ac.WriteLine ("Bugün hava berbat, fakat tahminlere göre yarın hava güneşli ve sıcaklık 21 derece civarında seyredecek...");
    ac.WriteLine ("Yeni çalışmalar için Ahmet aranacak. Eğer yeni üründen de bahsedersen iyi olur. <u>Aramayı unutma!</u>");
    ac.WriteLine ("Counter Strike için yapılan Botlar ne sıkıcı. Artık daha akıllı Bot yapmalılar. Devamlı <u>takılıyorlar</u>.");
    ac.WriteLine ("cs_dust oynamayalım. Kaç saattir bu haritayı ezberledik. Biraz da cs_estate olabilir.");
    ac.WriteLine ("<a href='http://www.tbagriyanik.cjb.net'>Sitemi ziyaret edin.</a> Hem orada tüm çalışmalarımı bulabilir ve
haberlar olabilirsiniz");
    ac.WriteLine ("Ütüğü <i>fişten</i> çektin mi? Yoksa ne olabilir tahmin et!");
    ac.WriteLine ("<hr>Bu hafta Yerli Malları Haftası. Kutlamayı unutmayın...<hr>");
    ac.WriteLine ("Programcılık en zevkli bilgisayar uğraşdır. <font color=green>Meslek</font> olarak da en güzeldir.");
    ac.WriteLine ("Ne desem bilmem ki! Kendinize iyi bakın...");
    ac.WriteLine ("Hile yapmadan oynanan oyun en zevklisidir...");
    ac.WriteLine ("Siz olsanız hangi ülkeye tatile gitmek isterdiniz? Madagaskar mı?... <font color=red>:</font>");
    ac.WriteLine ("<font color=orange>oleee </font>Ne güzel yaa, her yazdığım çıkabiliyor");
    ac.WriteLine ("<font color=yellow>İstedğin kadar yaz...</font>");
    ac.WriteLine ("Artık YETER");
    ac.WriteLine ("Bu son satirdir. Üsttekiler ekrana rastgele gelebilir...");
    ac.Close();
    document.write ("Dosya c:\yazi.txt olarak oluşturuldu... Birazdan çalışmaya başlayacak");
    window.setTimeout("window.navigate('alt.htm')", 3000); //3 sn sonra kendini çağırır
  }
}

function yaz()
{ yazi = Math.round((Math.random()*(i-1)))
  document.open()
    document.write("<b><MARQUEE BEHAVIOR=SCROLL DIRECTION=up LOOP=-1 SCROLLDELAY=0
SCROLLAMOUNT=2 WIDTH=180 HEIGHT=180>",arraym[yazi],"</MARQUEE></b>")
    document.close() //görüldüğü gibi asıl kaydırma komutu MARQUEE'dir
    window.setTimeout("window.navigate('alt.htm')", 7000);
}
dosyaAc(); //ana komut
//-->
</SCRIPT>
</body>
</html>
```

Cookie ise "Windows\Cookies" klasöründe saklanan metin tabanlı boyutu küçük, kullanıcının gezdiği sitelerin geçici bilgilerinin saklandığı dosyalardır. Web sitelerini dolaşırken sadece insanlar ana bilgisayarlardan bilgi çekmezler, sunucu da istemciden bilgi talep edebilir. Her bağlantıda bu bilgilerin sabit kalabilmesi bu küçük metin dosyaları sayesinde olur. Netscape firmasının etkileşim için çıkardığı bu sistem ile sitelerin davranışı kullanıcının özelliklerine göre değişebiliyor. Başka siteler yetkileri olmadan diğer cookie'leri okuyamazlar. Internet Explorer 6.0 ve üzeri web tarayıcıları bir siteye girildiğinde kendiliklerinden cookie kontrolü yapıyorlar. Bunu durum çubuğundan anlayabilirsiniz. Böylece zararlı cookie tutulması önleniyor. Adaware adlı bir programla da reklam kaynaklı bilgi sızdıran dosyaların silinmesini sağlayabilirsiniz. Spybot – Search&Destroy programı da Türkçe, bedava ve internetten güncellenebiliyor.

*.asp dosyalarımız içinde bu bilgileri yazma ve okuma komutlarını kullanabiliriz. "Ornekler\Diger\Java\cookie.html" ile JavaScript kullanarak cookie, yanda kayan menü, fare takipçisi gibi değişik numaralar elde ediyoruz. JavaScript modüler fakat VBScript kullanan *.asp daha basit. Kopyala yapıştır ile script kısımlarını kendi web sayfalarınıza ekleyebilirsiniz. "Ornekler\Diger\Java\hesapmakinesi.htm" ile de yine Java'nın gücü belli oluyor.

Bir cookie'nin en fazla 4 KB olabileceğini düşünerek şu şekilde değerler kaydedilir:

```
<%
Response.Cookies("cerezimiz")("a")= "Deneme 1"
Response.Cookies("cerezimiz")("b")= "Deneme 2"
Response.Cookies("cerezimiz")("c")= "Deneme 3"
Response.Cookies("cerezimiz").Expires= "November 30, 2001"
Response.Cookies("cerezimiz").HasKeys
%>
```

Aynı değişken adlarını kullanarak, tekrar okumak için de şöyle yapabiliriz:

```
<%
a=Request.Cookies("cerezimiz")("a")
b=Request.Cookies("cerezimiz")("b")
c=Request.Cookies("cerezimiz")("c")
Response.Write a
Response.Write b
Response.Write c
%>
```

Eğer `<%= Request.Cookies("cerezimiz") %>` yapsaydık tüm değişkenler ekrana gelirdi:

C=Deneme+3&B=Deneme+2&A=Deneme+1

`<%= Request.Cookies("cerezimiz")("a") %>` ekrana sadece tek bilgi gelir:

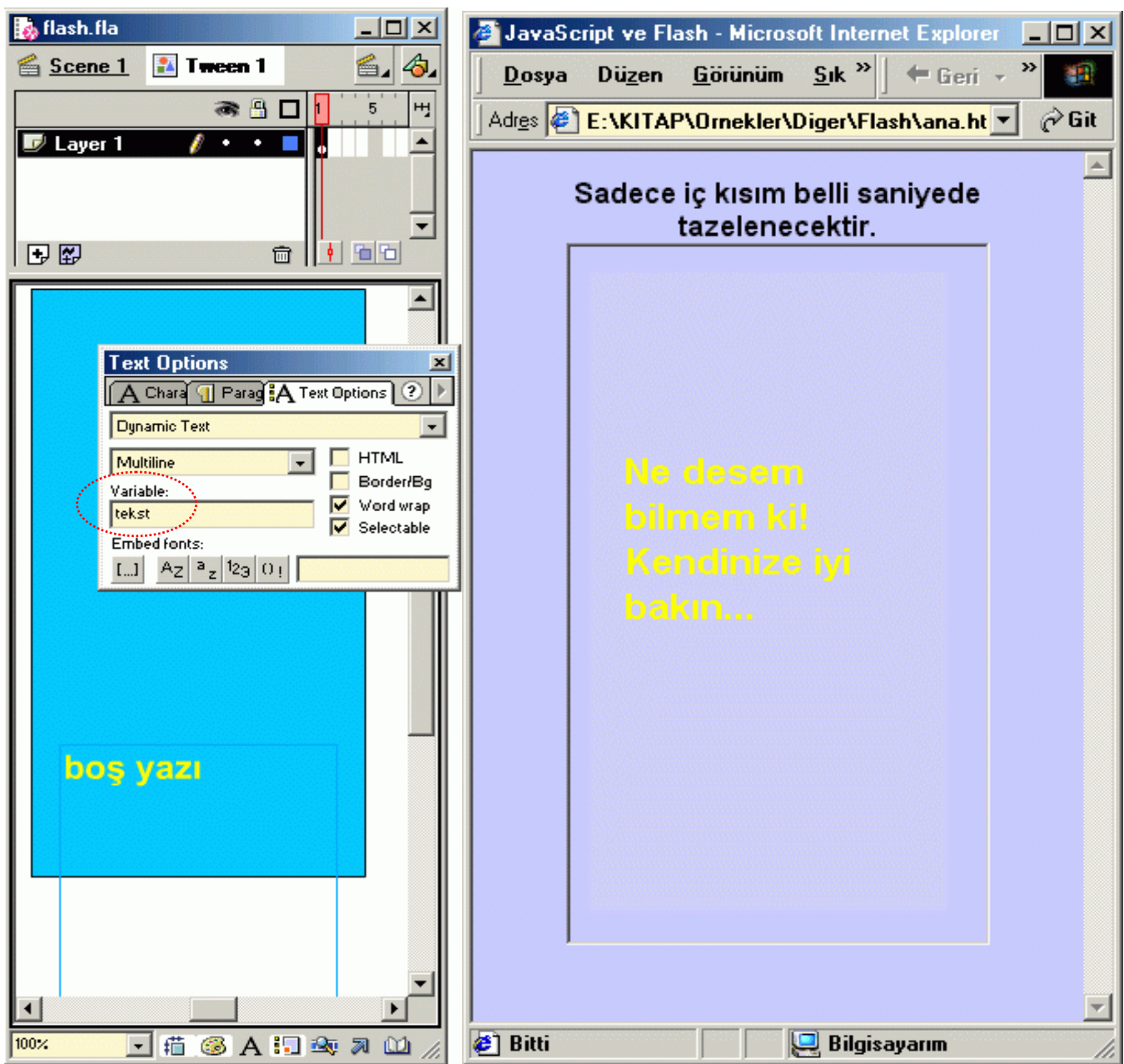
Deneme 1

10. JavaScript ve Flash (rasgele yazı kaydırılması)

(Ornekler\Diger\Flash\ana.htm)

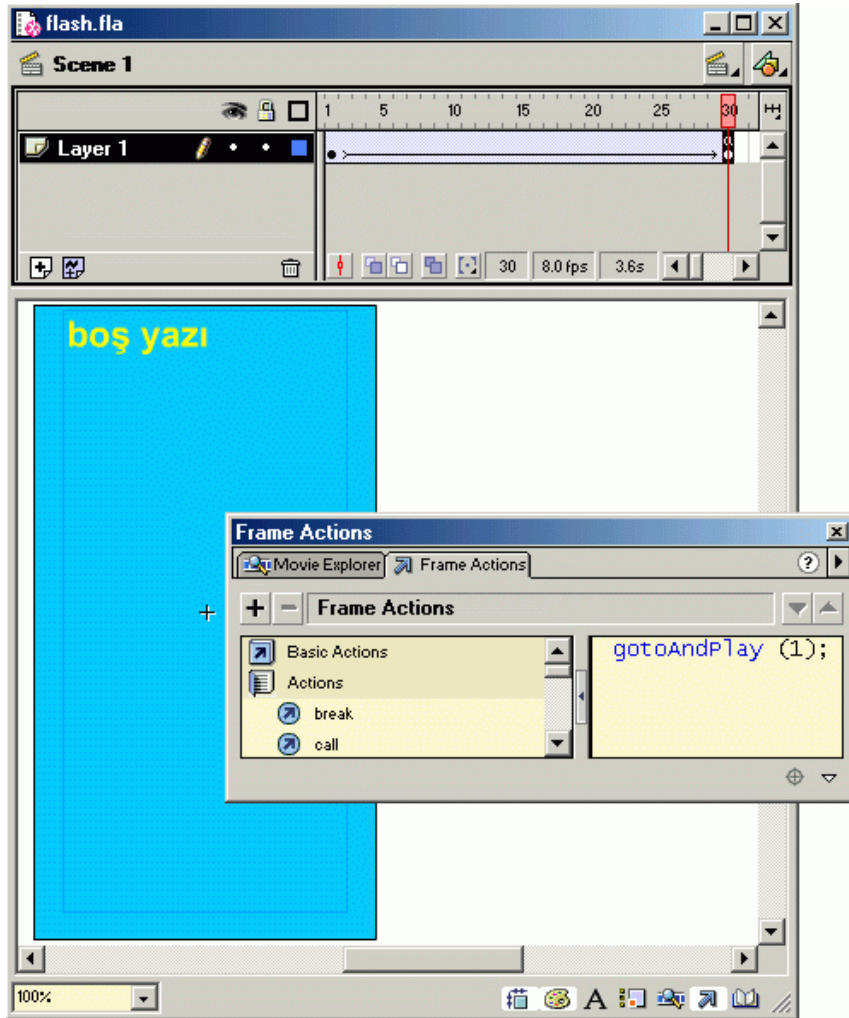
Üstteki kayan yazı örneğinin Macromedia Flash ile benzerini oluşturabiliriz. Sonuçta flash fla, flash.swf, ana.htm, alt.htm ve yazı2.txt adlı dosyalar elde edeceğiz. alt.htm gibi bir sayfaya asıl kodları yazıyoruz. Fakat doğrudan olarak ana sayfaya yapmayışımızın bir nedeni de, her tazelemeye sorulan güvenlik sorgulamasını, ilk açılışta sorulan tek soruya indirmektir. Bu arada üstteki örnekteki yazı.txt içindeki artık kullanamayacağımız HTML kodlarını kaldırıp, yazı2.txt gibi ikinci bir yazı dosyası yapmak zorunda kalıyoruz. Çünkü Flash metin içindeki HTML kodlarını yorumlamayıp aynen yazıyor. Zaten Flash'ın efektleri HTML'den çok daha yetenekli.

Önce Flash programı ile yukarı doğru kayan bir yazı hazırlıyoruz. Bir Text nesnesini sahneye ekliyoruz. Character – karakter özelliklerini Arial Tur, 21 punto, kalın ve sarı renk yapıp, Text Options kısmında da Dynamic Text (dinamik yazı), Multiline (çok satırlı) ve Wordwrap (kelime kaydır) özelliklerini değiştiriyoruz. Resimdeki gibi düzenlemeleri yapabilirsiniz. Variable – değişken adına **tekst** demeyi unutmayın. JavaScript ile bu değişkene değer yollayacağız.



Resim 15 – Flash tasarım anı ve sonuç ekranı

Yazı kısmı hazırlanınca ilk frame – çerçeve (zaman çizgisinde 1 numaralı çerçeve) üzerine sağ tıklayıp “Create Motion Tween” komutu veriyoruz. Son çerçeve için layer1 üzerinde 30’ncu kare tıklanıp F6 tuşu ile keyframe – anahtar çerçeve ekliyoruz. Yazının yukarı doğru gitmesi için bu son çerçevede yazıyı klavyeden Shift tuşuna (yanlara kaymaması için) basarken yukarı taşıyın. Yine 30’ncu çerçeve sağ tıklanıp Action – hareket kod kısmına gotoAndPlay (1); diyoruz. Aslında zaten kendiliğinden otomatik olarak loop – tekrar aynı hareketleri yapıyor. Bunu yapmasanız da tekrar başa döner demek istiyorum.



Resim 16 – Motion Tween ve Action Script yazmak

Resim 15’de görebileceğiniz 8.0 fps yazısının olduğu yere çift tıklayıp 8 fps – frames per second (saniyedeki kare sayısı), Width 200 px – pixel, Height 370 px yaparak hız ve boyut ayarlarımızı yapabiliriz. Aynı menüdeki Background Color ile mavi gibi bir renk seçebilirsiniz, fakat arka plan rengini alt.htm dosyasında değiştiriyoruz. Ctrl+Enter tuşarı ile hem flash.swf dosyamızı elde ediyor, hem de önizleme yapabiliyoruz. Sadece Enter ile de tasarım ve çalışmayı aynı anda görebiliriz. “File*Publish Settings...” ile *.htm ve *.swf seçili olursa, “File*Publish Shift+F12” dediğimizde alt.htm içinde kullanacağımız <object> tag – etiket kodlarının hazır olarak gelmesini sağlayabiliriz.

ana.htm ve alt.htm kodları şu şekilde not defteri ile hazırlanabilir:

```
<html> <head>
<title>JavaScript ve Flash</title>
<meta http-equiv="Content-Type" content="text/html; charset=""/></head>
<body bgcolor="#CCCCFF">
<center>
<font face="Arial, Helvetica, sans-serif"><b>Sadece
iç kısım belli saniyede tazelenecektir.<br>
<iframe width="235" height="405" src="alt.htm" ></iframe>
</b> </font>
```

```
</center>
</body>
</html>
```

alt.htm:

```
<html><head>
<title>JavaScript ve Flash</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<meta http-equiv="Content-Language" content="tr">
</head>
<body bgcolor="#CCCCFF">
<OBJECT id='flash' classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0" WIDTH=200
HEIGHT=370><PARAM NAME=movie VALUE="flash.swf"> <PARAM NAME=quality VALUE=high> <PARAM NAME=bgcolor
VALUE=#CCCCFF> <EMBED name="flash" src="flash.swf" quality=high bgcolor=#CCCCFF WIDTH=200 HEIGHT=370
TYPE="application/x-shockwave-flash"
PLUGINSPAGE="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash"></EMBE
D><OBJECT>
<SCRIPT LANGUAGE = "JavaScript1.2">
<!--Javascript kodunu eski sürüm browserlardan saklayalım
var arraym = new Array()
var dosya, ac, i, yazi;
var ForReading = 1; ForWriting = 2;
function dosyaAc()
{ dosya = new ActiveXObject("Scripting.FileSystemObject");
  if (dosya.FileExists("c:\\yazi2.txt"))
  {ac = dosya.OpenTextFile("c:\\yazi2.txt", ForReading);
   i = 0;
   document.open()
   while(!ac.AtEndOfStream)
   {arraym[i] = ac.ReadLine();
    i++;
   }
   yaz();
  }
  else
  {
    ac = dosya.CreateTextFile("c:\\yazi2.txt", ForWriting);
    ac.WriteLine ("Bugün hava berbat, fakat tahminlere göre yarın hava güneşli ve sıcaklık 21 derece civarında seyredecek...");
    ac.WriteLine ("Yeni çalışmalar için Ahmet aranacak. Eğer yeni üründen de bahsedersen iyi olur. Aramayı unutma!");
    ac.WriteLine ("Counter Strike için yapılan Botlar ne sıkıcı. Artık daha akıllı Bot yapmalılar. Devamlı takılıyorlar.");
    ac.WriteLine ("cs_dust oynamayalım. Kaç saattir bu haritayı ezberledik. Biraz da cs_estate olabilir.");
    ac.WriteLine ("http://www.tbagriyanik.cjb.net sitemi ziyaret edin. Hem orada tüm çalışmalarımı bulabilir ve haberdar
olabilirsiniz");
    ac.WriteLine ("Ütüyü fişten çektin mi? Yoksa ne olabilir tahmin et!");
    ac.WriteLine ("Bu hafta Yerli Malları Haftası. Kutlamayı unutmayın...");
    ac.WriteLine ("Siz olsanız hangi ülkeye tatile gitmek isterdiniz? Madagaskar mı?... :)");
    ac.WriteLine ("olee Ne güzel yaa, her yazdığım çıkabiliyor");
    ac.WriteLine ("İstediğin kadar yaz...");
    ac.WriteLine ("Artık YETER");
    ac.WriteLine ("Bu son satırdır. Üsttekiler ekrana rastgele gelebilir...");
    ac.Close();
    window.setTimeout("window.navigate('alt.htm')", 100);
  }
}

function yaz()
{
  yazi = Math.round((Math.random()*(i-1)));
  window.document.flash.SetVariable("tekst",arraym[yazi]);
  window.setTimeout("window.navigate('alt.htm')", 7000);
}
dosyaAc(); //ana komut
<!-->
</SCRIPT>
</body> </html>
```

window.document.flash.SetVariable("tekst",arraym[yazi]); komutu sayesinde Flash içindeki herhangi bir nesneye değer atabiliyoruz. flash kelimesi bizim oluşturduğumuz flash.swf dosyasının adıdır ve alt.htm'nin üst satırlarındaki <object> kısmındaki yerde id – identification (kimlik) ve embed kısmında da name – ad ile bunu belirtmeliyiz.

Flash içine komut yollamanın yanında getURL(); ile dışarıya yani Internet Explorer penceresine de komutlar yollayabiliriz. Bu web sayfası içindeki bir JavaScript fonksiyonu da olabilir. Nesne veya düğmenin Action komutlarına şu örnekleri yazabilirsiniz:

getURL ("JavaScript:dosyaAc();");

Bir düğmenin kod kısmına şunlar yazılarak web sayfası kapatılabilir:

```
on (release){
getURL("JavaScript:top.close();"); //pencere kapatılmadan önce Internet Explorer onay istiyor
}
```

Başka bir düğmenin kod kısmına şunlar yazılarak ayrı bir pencere açılabilir:

```
on (release){
getURL ("sayfa2.html", "blank");
}
```

Flash içindeki başka bir nesnenin özelliklerine şu şekilde ulaşılabilir:

```
setProperty ("nesneadi", _x, 250); //nesneadi 'nin x koordinatı değiştirilir
setProperty ("nesneadi ",_x, getProperty("nesneadi ", _x)-artim);
```

Verilen hedefte(bir movie clip olabilir) örneğin pop menüler gibi kontroller yapmak için:

```
tellTarget ("hedefadi")
{
gotoAndPlay(5); //burada başka işlemler de yapılabilir
};
```


11. JavaScript ile arama (*.html içinde arama)

(Ornekler\Diger\Java\arama.htm)

Sayfa içinde belli kelimeyi aramak ve kaç adet olduğunu bulmak için bu kodları kullanabiliriz. Netscape için farklı komut kullanımı gerekebilir. Netscape'e özel kod ile uğraşmama nedenim dünyada büyük bir çoğunluğun bu browser - tarayıcıyı kullanmaması. Şu anda çoğu kişi Internet Explorer 5 ve üzerini doğal olarak kullanıyorlar. İsterseniz programı geliştirerek bulunan satırları başka bir sayfada tablo halinde listeleyebilirsiniz.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<title>Arama</title>
</head>
<body bgcolor="#00ccff">

<font face="Arial">The help is in the "Help" subdirectory, in HTMLHelp
format (check the "download" section if you are using Win95 and need
HTMLHelp support). It is regularly updated, but GLScene is currently growing
faster than the help file.</font>

<form name="form1">
<p align="center"><input name="string"><br>
  <input type="button" value="Ara/Sonraki" name="B1" onclick="bul();">
  <input type="reset" value="Temizle" name="B2"></p>
  <input type="text" name="textfield" value="TB" maxlength="0" size="1">
</form>

<script language="JavaScript1.2">
  var n = 0;
  function bul()
  {
    var txt,i,tamam;
    str = document.form1.elements[0].value;
    if (str!="")
    {
      txt = document.body.createTextRange();
      for (i=0;i<= n &&(tamam=txt.findText(str))!=false;i++)
      {
        txt.moveStart("character",1);
        txt.moveEnd("textedit");
      }
      if (tamam)
      {
        txt.moveStart("character",-1);
        txt.findText (str);
        txt.select();
        txt.scrollIntoView();
        n++;
      }
      else
      {
        if (n>0)
        {
          alert("Toplam "+n+" adet bulundu"); n=0;
        }
        else alert (str+" bulunamadı...");
      }
    }
  }
</script>
</body>
</html>
```

12. ASP ile misafir defteri (*.asp ve *.mdb veritabanı)

(Diger\ASP\myasp\myasp1.zip ve myasp2.zip)

Temel olarak *.asp dosyalar olan Active Server Pages – etkin sunucu sayfaları ile, internet veya intranet üzerinde veritabanı ve güvenlik gerektiren sayfaları oluşturabilirsiniz. PHP ve Perl – Practical Extraction and Report Language gibi C diline benzeyen diğer CGI – Common Gateway Interface programlama dilleri ile de daha profesyonel sayfalar yapmak mümkündür. Ayrıca Macromedia Flash gibi animasyonlu web sayfaları rahatlıkla ASP ile ilişkilendirilebilir. Genellikle tüm siteyi *.asp olarak hazırlamak yerine kullanıcı logon – girişi, misafir defteri, forum ve benzeri anket gibi analiz edilebilen verilerin olduğu internet sayfalarında kullanabiliriz. brinkster.com gibi bedava ASP desteği veren siteler ile de yaptığınız sayfaları deneyebilirsiniz, ya da ücretli olarak edindiğiniz host - sunucu üzerinde işlevleri test edebilirsiniz. Aslında çoğu sunucu NT/2000/2003 ve Unix tabanlıdır. Bu işletim sistemleri doğal olarak CGI gibi hizmetleri içerirler. Brinkster üyeliği sonrasında db klasörüne attığınız *.mbd için yazma hakkı olmadığına dair bir hata mesajı alabilirsiniz. Live Support ile bu isteğinizi belirtirseniz, veritabanınıza kayıt girme ve silme gibi işlemleri de yapabilirsiniz.

CGI dosyaları istemci yani kullanıcıların web sayfasını sunucudan istemesi ile çalışmaya başlar. Kullanıcı penceresinde *.htm dosyayı gördüğü halde aslında bu sayfa sunucu üzerinde çalışan *.asp dosyasıdır. Form bilgileri ve seçeneklere göre *.asp dosyanın son işlenmiş *.htm hali farklı olabilir. CGI sayesinde istemci, sunucuya komut yollar ve onu programlar. Fakat onlarca kişi aynı anda sunucudan hizmet bekleyince sunucu gittikçe yavaşlamaya başlayabilir. JavaScript istemci tarafında çalışırken, ASP sunucu tarafında çalışır. Böylece web browser – tarayıcı programı ne olursa olsun *.htm kodunu kullanıcı rahatlıkla görecektir. Bu arada *.asp dosyalarındaki kodların aslında VBScript olduğuna da dikkatinizi çekirim. Bunun anlamı JavaScript ile de *.asp dosya oluşturulabileceğidir. VBScript Netscape tarayıcısı tarafından tanınmadığı halde *.asp kodlarımız sunucu tarafında çalışıp *.htm halini yollayacaktır. Kullanıcı sizin *.asp kısmındaki kodları asla göremeyecek ve sadece özet bilgi gibi önüne gelen *.htm ile yetinecektir. Orijinal *.asp kodu örneğin 3KB iken, sonuç *.htm dosyası 1KB'dan MB seviyesine kadar büyük dosya olabilir. Çünkü sonuç en azından basit bir hata mesajı olabilir veya büyük bir veritabanının ekrana tablolanması da olabilir.

Kendi bilgisayarımızda *.asp sayfaları oluşturmamız için gerekenler sadece not defteri, Microsoft Access ve Microsoft Personal Web Server (PWS – Kişisel Web Sunucusu) programlarıdır. PWS programı Win98 CD'sinde "Tools\Add-ons\pws" dizininde bulabilirsiniz. Kurulumdaki root – kök dizin sorulduğunda "C:\Belgelerim\My Webs" gibi bir klasör adını belirledikten sonra, saat simgesinin yanındaki basit menülere sahip olan PWS programına ulaşabilirsiniz. NT ve üzeri için ise Internet Information Services – IIS adlı programı kurmanız gerekir.

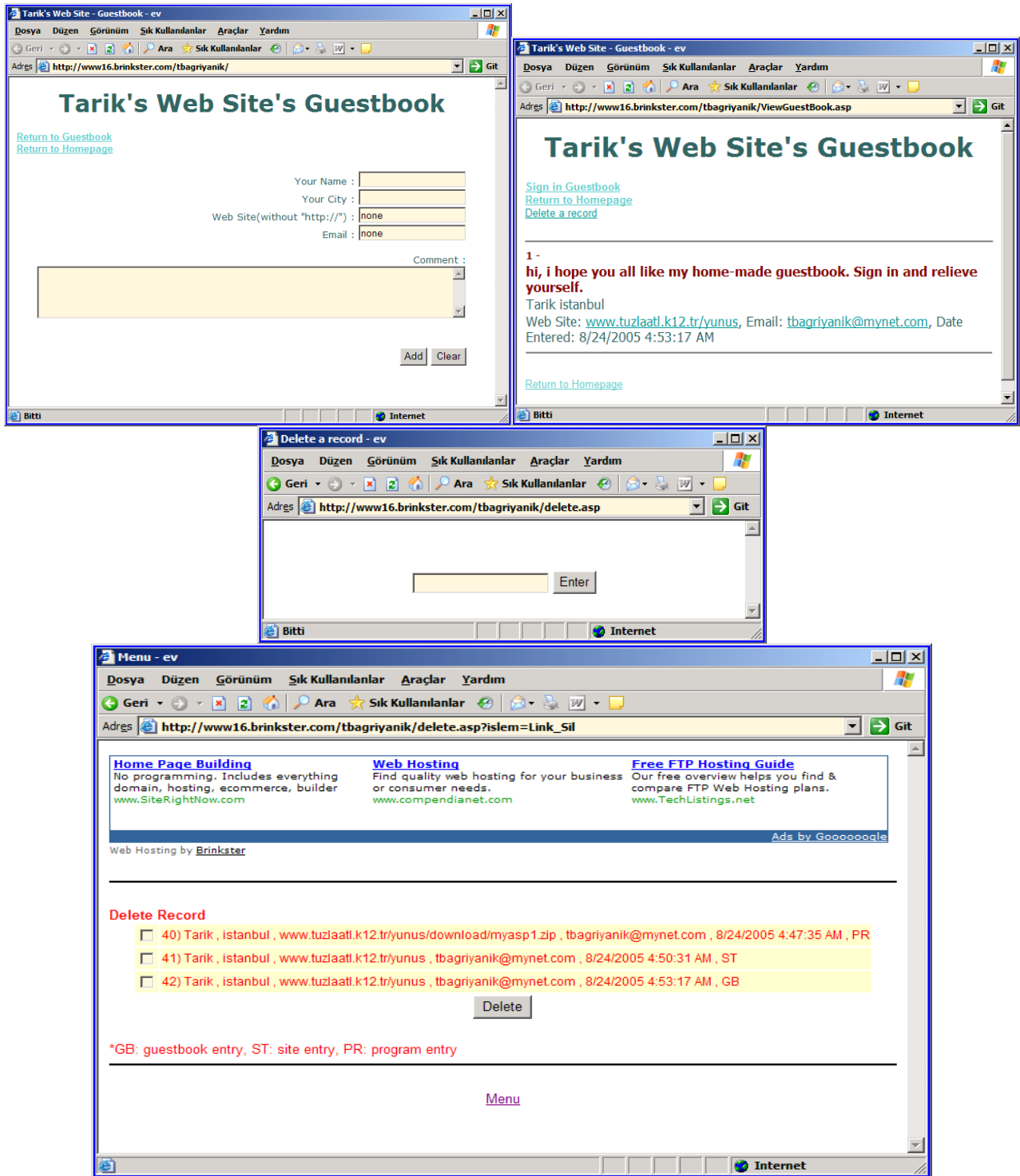
Internet Explorer adres çubuğundan internete bağlanmadan, "<http://localhost>" denildiğinde, artık sizin bilgisayarınızdaki default.asp, index.htm gibi sayfalarımızı çalıştırıp *.asp uzantılı dosyaların canlı hale geldiğini görebilirsiniz. İsterseniz "Denetim Masası*Ağ" kısmından bilgisayar adını değiştirerek "<http://server>" gibi de adlandırabilirsiniz. Perl için *.pl uzantılı dosyaların aktif hale gelmesi ActivePerl adlı program sayesinde olur.

MS Access ile "My Webs\db" isimli bir dizin içine guestbook.mdb adlı basit bir veritabanı oluşturacağız. tblContacts adlı bir Table – tablo ekleyerek alan adlarını şu şekilde yapıyoruz:

ContactID	AutoNumber	
FirstName	Text	Size 30
LastName	Text	Size 30
Phone	Text	Size 100
Email	Text	Size 50
Comments	Note	
DateEntered	Date/Time	Default Value =NOW()
Type	Number	

Bu projede 3 farklı alan birbirine benzer şekilde birleştirilmiş. Misafir defteri, program ekleme, site ekleme modülleri. Arabirimleri aynı fakat grupları(Type) farklıdır. Yani aynı veritabanına kaydolurken, kaydın türüne göre filtre amacı ile numara veriyoruz. Mesela misafir defteri 1, program ekleme 2 ve site ekleme 3.

index.html misafir defteri için, index1.html program ekleme için ve son index2.html de site ekleme için gereken ana sayfadır. Tek ana sayfa da yapılabilirdi. Kullanıcılar üye yapılarak, kendi misafir defteri de yapma imkanı olabilirdi. Böylece misafir defteri hizmeti de verebilirdik.



Resim – Asp projemizin çalışır haldeki ekran görüntüleri

Ana menü yani veri giriş ve listeleme için, "My Webs\index.html" içine sayfa oluşturuyoruz:

```
<html><title>Tarik's Web Site - Guestbook</title>
<body bgcolor="#FFFFFF" text="#336666" link="#009999" alink="#00FFFF" vlink="#66CCCC">
<font face="Verdana, Arial, Helvetica, sans-serif"><h1 align="center">Tarik's Web Site's Guestbook</h1>
<a href="ViewGuestBook.asp"><b><font size="2" face="Arial, Helvetica, sans-serif">Return
to Guestbook</font></b></a><font size="2" face="Arial, Helvetica, sans-serif"><b><br>
<A HREF="http://www.yunus.projesi.com" target="_self">Return
to Homepage</A></b></font> </font>
<form name="form1" method="post" action="GuestBook.asp">
```

```

<table width="65%" border="0" height="44%" align="center">
<tr>
<td>
<p align="right"><font face="Verdana, Arial, Helvetica, sans-serif" size="2">Your
Name :
<input type="text" name="FirstName" maxlength="30">
<br>
Your City :
<input type="text" name="LastName" maxlength="30">
<br>
Web Site(without &quot;http://&quot;):
:
<input type="text" name="Phone" value="none" maxlength="100">
<br>
Email :
<input type="text" name="Email" value="none" maxlength="50">
<br>
<br>
Comment :
<textarea name="Comments" rows="5" cols="70"></textarea>
</font> <b> <font color="#009999">
<br>
</font></b><b><font color="#009999">
<br>
<input type="submit" name="Submit" value="Add">
<input type="reset" name="Submit2" value="Clear">
</font> </b> </p>
</td>
</tr>
</table>
</form>
</html>

```

Bilgileri index.html'den alıp kontrol ederek kaydeden dosyamız ise "GuestBook.asp" 'dir:

```

<% @ LANGUAGE="VBSCRIPT" %>
<%

```

Option Explicit

```

Dim rst
Dim strFirstName
Dim strLastName
Dim strPhone
Dim strEmail
Dim strComments

```

```

dim DatabasePath

```

```

Dim ado_OpenKeyset
Dim ado_LockOptimistic
Dim strConnect

```

```

function temizle(metin)
metin=replace(metin,chr(10),"") 'satirbasi
metin=replace(metin,chr(34),"") 'tirnak
metin=replace(metin,"'", "") 'tirnak
metin=replace(metin,"<","")
metin=replace(metin,">","")
temizle=trim(metin)
end function

```

```

'Connection String
DatabasePath = "db/guestbook.mdb"
strConnect= "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ= " & server.mappath(DatabasePath)
ado_OpenKeyset=1
ado_LockOptimistic=3

```

```

'get user input from the form fields
strFirstName = temizle(Request.Form("FirstName"))
strLastName = temizle(Request.Form("LastName"))
strPhone = temizle(Request.Form("Phone"))
strEmail = temizle(Request.Form("Email"))
strComments = temizle(Request.Form("Comments"))

'set up rst; create the recordset object
Set rst=server.createObject("ADODB.RECORDSET")
%>

<html>

<head>
<title>Tarik's Web Site - Guestbook</title>
<meta name="keywords" content="asp, active server pages, learning, beginners, newbie, asp, java, access 2000 2000 visual
basic vb resources MSAccess 97 microsoft programming web design web ">
</head>
<body bgcolor="#FFFFFF" text="#336666" link="#009999" alink="#00FFFF" vlink="#66CCCC">
<font face="Verdana, Arial, Helvetica, sans-serif" size="2">
<h1 align="center">Tarik's Web Site's
Guestbook</h1>
<p> <a href="index.html"><font face="Arial, Helvetica, sans-serif"><b>Sign
in Guestbook</b></font></a><b><font face="Arial, Helvetica, sans-serif"><br>
<a href="ViewGuestBook.asp">Return
to Guestbook</a><br>
<A HREF="http://www.yunus.projesi.com" target="_self">Return
to Homepage </A> <br>
</font></b> </p>

<%
If strFirstName="" or strLastName="" or strPhone="" or strEmail="" or strComments="" Then
    Response.Write "<font color=red>You did not enter all, please do.</font>"
Else
    rst.open "tblContacts", strConnect, ado_OpenKeyset, ado_LockOptimistic

    rst.AddNew
    rst("FirstName") = strFirstName
    rst("LastName") = strLastName
    rst("Phone") = strPhone
    rst("Email") = strEmail
    rst("Comments") = strComments
    rst("Type") = 1
    rst.Update
    Response.Write "<font color=blue>Thank you for your patience... " & strFirstName & " from " & strLastName &
"</font>"
    rst.Close
    set rst = Nothing
end if

%> </font>
</body>
</html>

```

Bilgileri ekrana listeleyen sayfa ise "ViewGuestBook.asp".

```

<%@ Language=VBScript%>
<%Option Explicit%>

<html>
<head>
<TITLE>Tarik's Web Site - Guestbook</TITLE>
<meta name="keywords" content=" asp, active server pages, learning, beginners, newbie, asp, java, access 2000 2000 visual
basic vb resources MSAccess 97 microsoft programming web design schedules office 2000">

```

```
</head>
```

```
<body bgcolor="#FFFFFF" text="#336666" link="#009999" alink="#00FFFF" vlink="#66CCCC">
<h1 align="center" msnavigation><font face="Verdana, Arial, Helvetica, sans-serif">Tarik's
Web Site's Guestbook</font><font face="Arial, Helvetica, sans-serif" size="2"></font></h1>
<p align="left"> <b> <A href="index.html"><font face="Arial, Helvetica, sans-serif" size="2">Sign
in Guestbook</font></a>
<font face="Arial, Helvetica, sans-serif" size="2"><br>
<a href="http://www.yunus.projesi.com">Return to Homepage</a>
<br></b>
<a href="delete.asp">Delete a record</a> </font></b>
</p>
<tr><td valign="top" width="1%"></td><td valign="top" width="24%"></td>
<td valign="top" msnavigation>
<hr noshade>
<font size="2"><%
Dim rst
```

```
dim DatabasePath
```

```
Dim ado_OpenKeyset
Dim ado_LockOptimistic
Dim strConnect
Dim strSQL
Dim kayitno
```

```
'Connection String
DatabasePath = "db/guestbook.mdb"
set strConnect = server.createobject("ADODB.Connection")
strConnect.Open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ= " & server.mappath(DatabasePath)
```

```
'I am using this sql instead of the usual recordset object to sort data in desc order
strSQL = "SELECT * FROM tblContacts " & _
"where type=1"
```

```
'Execute the SQL statement'
set rst = strConnect.execute(strSQL) 'this creates our recordset
```

```
kayitno = 0
%> </font>
<div align="left">
<table border="0" cellspacing="0" width="100%" bgcolor="#ffffff">
<%
'loop through the table
do while not rst.EOF
kayitno=kayitno+1
%>
<tr>
<td width="75%"><font color="#800000" size="2"><b><%Response.write kayitno & " - " %></b></font>
</td>
</tr>
<tr>
<td width="75%"><font color="#800000"><b><%Response.write rst("Comments")%></b></font>
</td>
</tr>
<tr>
<td width="100%"><%Response.write rst("FirstName") & " " & rst("LastName") %>
</td>
</tr>
<tr>
<td width="100%"> <%
if rst("Phone")<>"none" then
Response.write "Web Site: "
Response.write "<a href='http://'&rst("Phone")&'" target='_blank'>"& rst("Phone")&"</a>, "
end if
if rst("Email")<>"none" then
```

```

    Response.write "Email: "
    Response.write "<a href='mailto:"&rst("Email")&">" & rst("Email")&"</a>,"
    end if
    Response.write "Date Entered: "& rst("DateEntered")
    %> </td>
</tr>
<tr>
    <td width="100%">
        <hr noshade>
    </td>
</tr>
<%
    rst.MoveNext
    loop
'clean up
    rst.Close
    set rst = Nothing
%>
<tr>
    <td valign="baseline" colspan="3" width="100%">
        <p align="left">&nbsp;</p>
    </td>
</tr>
</table>
<font face="Arial, Helvetica, sans-serif" size="2">
<a href="http://www.yunus.projesi.com">Return to Homepage</a>
</div>
<font size="2">&nbsp;</font>
</body msnavigation>
</HTML>

```

Kayıt silme kısmı "delete.asp" :

```

<% Response.Buffer = True
Response.Expires = 0 %>
<html>
<head>
<title>Menu</title>
<style type="text/css">
<!--
body,td,th {
    font-family: Arial, Helvetica, sans-serif;
}
.style32 {font-size: small}
.style33 {
    font-size: x-small;
    font-weight: bold;
}
.style34 {font-size: x-small}
.style35 {font-size: 12px}
-->
</style>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9"></head>

<body bgcolor="#FFFFFF" text="#FF0000">
<span class="style5"><font size="2">
<%

veritabani = "Driver={Microsoft Access Driver (*.mdb)}; dbq=" & server.mappath("db/guestbook.mdb")
set bag = Server.CreateObject("adodb.connection")
set ks = Server.CreateObject("adodb.recordset")

%>
<%
'#####

```

```
dogruparola = "12345678" ' Kendiniz yeni bir parola belirleyiniz.CHOOSE PASSWORD
#####
```

```
If request("parola") = dogruparola or session("parola") = dogruparola then
    If request("islem") = "Ana" then
        Call Ana
        session("parola") = dogruparola
    ElseIf request("islem") = "Yeni_Link" then
        Call Yeni_Link
        session("parola") = dogruparola
    ElseIf request("islem") = "Link_Duzenle" then
        Call Link_Duzenle
        session("parola") = dogruparola
    ElseIf request("islem") = "Link_Sil" then
        Call Link_Sil
        session("parola") = dogruparola
    ElseIf request("islem") = "KayitSil" then
        Call KayitSil
        session("parola") = dogruparola
    ElseIf request("islem") = "KayitEkle" then
        Call KayitEkle
        session("parola") = dogruparola
    ElseIf request("islem") = "DuzenleKaydet" then
        Call DuzenleKaydet
        session("parola") = dogruparola
    ElseIf request("islem") = "cikis" then
        session("parola") = ""
        response.redirect (Request.ServerVariables("HTTP_REFERER"))
    ElseIf request("islem") = "Link_DuzenleAna" then
        Call Link_DuzenleAna
        session("parola") = dogruparola
    Else
        Call Ana
        session("parola") = dogruparola
    End If
Else
    Call Parola
End If
```

```
%>
<%
'=====
%>
<% SUB Ana %>
<!-- Ana Menü -->
</font>
</span>
<table border="0" width="141" cellpadding="0" cellspacing="0" align="center">
  <tr bgcolor="#DFDFFF">
    <th width="141"><div align="center"><span class="style7"> Menu</span></div></th>
  </tr>
  <tr bgcolor="#FFFFCA">
    <td> <div align="left">
      <ul>
        <li><span class="style9"><font size="2"><a href="?islem=Link_Sil" title="Record
delete">DELETE</a></font></span></li>
      </ul>
    </div></td>
  </tr>
  <tr bgcolor="#FFFFCA">
    <td> <div align="left">
      <ul>
        <li><span class="style11"><font size="2"><a href="?islem=Yeni_Link" title="New record">ADD</a></font> </span></li>
      </ul>
    </div></td>
  </tr>
```



```

<tr bgcolor="#FFFFCA">
  <td><div align="left">
    <ul>
      <li>
        <span class="style13"><font size="2"><a href="?islem=Link_Duzenle" title="Update a
record">UPDATE</a></font></span></li>
      </ul>
    </div></td>
  </tr>
<tr bgcolor="#FFFFCA">
  <td><div align="center"></div></td>
</tr>
<tr bgcolor="#FFFFCA">
  <td><div align="center"><span class="style17"><a href="?islem=cikis" title="Quit">Logout</a></span></div></td>
</tr>
</table>
<span class="style18"><font size="2">
<!-- Ana Menü -->
<% END SUB ' Ana %>
<%
'=====
%>
<%
SUB Parola
%>
<!-- Parola -->
</font>
</span>
<p align="center">
<form action="delete.asp" method=post name="yeni"><span class="style20">
  <font size="2">
    <input type="password" name="parola">
    <input type="submit" value="Enter">
  </font>
</span></form>
<span class="style21"><font size="2">
<!-- Parola -->
<%
END SUB ' Parola
%>
<%
SUB Link_Sil
%>
<!-- site_name Sil-->
<%
bag.open veritabani
sql = "select * from tblContacts order by DateEntered"
set kayitlar = bag.execute(sql)
%>
</font>
</span>
<form action="delete.asp" method="post" name="silmeFormu" id="silmeFormu">
<span class="style22"><font size="2">
<input type="hidden" name="islem" value="KayitSil">
</font>
</span>
<hr noshade color=black>
<span class="style22"><br>
<span class="style33">Delete Record
</span></span>
<table align="center">
  <% do while not kayitlar.eof %>
    <tr bgcolor="#FFFFCA">
      <td class="style32">
        <input name="<%=kayitlar("ContactID")%>" type="checkbox" value="sil">
      </td>

```

```

<td class="style35">
  <%=kayitlar("ContactID")%> <%=Left(kayitlar("FirstName"),250)%>
  , <%=Left(kayitlar("LastName"),250)%>
  , <%=Left(kayitlar("Phone"),250)%>
  , <%=Left(kayitlar("Email"),250)%>
  , <%=Left(kayitlar("DateEntered"),250)%>
  ,
  <%if kayitlar("Type")=1 then %>
  GB
  <% end if
if kayitlar("Type")=2 then %>
  ST
  <% end if
if kayitlar("Type")=3 then %>
  PR
  <% end if %>
</td>
</tr>
<%
kayitlar.movenext
loop
%>
<tr>
<td colspan=2 align=center class="style32">

  <input name="" type="submit" onClick="alert('Process done');" VALUE="Delete">
</td>
</tr>
</table>
<span class="style32"><br>
<span class="style34">*GB: guestbook entry, ST: site entry, PR: program entry</span></span>
<hr noshade color=black>
</form>
<p align=center><span class="style23"><font size="2"><a href="?islem=Ana">Menu</a> <%
kayitlar.close
set kayitlar = nothing
bag.close
set bag = nothing
%>
  <!-- site_name Sil-->
  <%
END SUB ' Link_Sil
%>
  <%
'=====
%>
  <%
SUB KayitSil
%>
  <!-- Kayyt Sil -->
  <%
bag.open veritabani
set ks1 = Server.CreateObject("ADODB.Recordset")
set ks2 = Server.CreateObject("ADODB.Recordset")
sql1 = "Select * from tblContacts"

ks1.open sql1, bag, 0, 1

do while not ks1.eof
sno = ks1("ContactID")
sno = cstr(sno)
if request.form(sno) = "sil" then

sql2 = "delete from tblContacts Where ContactID = " & sno
ks.open sql2, bag, 0, 1
end if

```

```
ks1.MoveNext
loop
```

```
ks1.Close
set ks1 = Nothing
set ks2 = Nothing
bag.close
set bag = Nothing
```

```
response.redirect (Request.ServerVariables("HTTP_REFERER"))
%>
<!-- Kayyt Sil -->
<%
END SUB ' KayitSil
%>
<%
'=====
%>
<% SUB Yeni_Link %>
</font>
</span><form action="delete.asp" method="post" name="form1yeni">
<table width="100%" border="0" height="44%" align="center">
<tr>
<td>
<p align="left"><span class="style24"><font size="2"> <strong>New Record</strong> <br>
Name <br>
<input type="text" name="FirstName" maxlength="30">
<br>
City<br>
<input type="text" name="LastName" maxlength="30">
<br>
Link(without &quot;http://&quot;)<br>
<input name="Phone" type="text" value="none" size="50" maxlength="100">
<br>
Email<br>
<input type="text" name="Email" value="none" maxlength="50">
<br>
Comment<br>
<textarea name="Comments" rows="5" cols="70"></textarea>
<font> <b><font color="#009999"> <br>
</font></b></span>
<label>
<span class="style35">
<input name="Type" type="radio" value="1" checked>
GuestBook</span></label>
<span class="style35"><br>
<label>
<input type="radio" name="Type" value="2">
Site</label>
<br>
<label>
<input type="radio" name="Type" value="3">
Program</label>
<br>
</span> <span class="style24"><b><font color="#009999">
<input name="Submit" type="submit" onClick="alert('Process started');" value="Add">
</font></b><font color="#009999">
<input type="reset" name="Submit2" value="Reset">
<input type="hidden" name="islem" value="KayitEkle">
</font> </b> </span></p> </td>
</tr>
</table>
</form>
<p align=center><span class="style26"><font size="2"><a href="?islem=Ana">Menu</a> <%
'=====
```

```

%>
<% END SUB ' Yeni_Link %>
<%
'=====
%>
<%
SUB KayitEkle
%>
<%
first_name = Request.Form("FirstName")
last_name = Request.Form("LastName")
city_name = Request.Form("Phone")
email_name = Request.Form("Email")
comment_name = Request.Form("Comments")
t_name = Request.Form("Type")
if first_name = "" or last_name = "" or comment_name = "" then
response.write("<p align=center>Empty fields! Press Back button to retry.")
response.end
end if

bag.open veritabani
sql ="Select * from tblContacts"
ks.open sql, bag, 1, 3
ks.AddNew
ks("FirstName") = first_name
ks("LastName") = last_name
ks("Phone") = city_name
ks("Email") = email_name
ks("Comments") = comment_name
ks("Type") = t_name
ks.update
ks.close
set ks = nothing
bag.close
set bag = nothing
response.redirect("?islem=Ana")
%>
<% END SUB ' KayitEkle %>
<%
'=====
%>
<% SUB Link_Duzenle %>
<!-- site_name Düzenle -->
</font>
</span><p align="center">
<form action="delete.asp" method="post" name="duzenNo" id="duzenNo"><span class="style28">
<font size="2">
<input type="Hidden" name="islem" value="Link_DuzenleAna">
Record number to edit
<input type="text" name="ContactID" size="4">
<input type="submit" value="Edit">
</font>
</span></form>
<p align=center><span class="style29"><font size="2"><a href="?islem=Ana">Menu</a> <!-- site_name Düzenle -->
<% END SUB ' Link_Duzenle %>
<%
'=====
%>
<% SUB link_DuzenleAna %>
<!-- Haber Düzenle Ana -->
<%
sno = Request.Form("ContactID")
%>
<%
bag.open veritabani

```

```

sql = "Select * from tblContacts Where ContactID= "& sno
set kayitlar = bag.execute (sql)

%>
  <% if kayitlar.eof then
response.write("<tr><td>Not found!</td></tr>")
else
%>
  </font>
</span><form action="delete.asp" method="post" name="formDuzenle" id="formDuzenle">
  <tr>
    <td valign="top"><span class="style30"><font size="2"><b> Editing
      Record Number</b> <%=kayitlar("ContactID")%><br>
      Name<br>
      <input type="text" name="FirstName" size="30" value='<%=kayitlar("FirstName") %>'>
      <br>
      City<br>
      <input type="text" name="LastName" size="30" value='<%=kayitlar("LastName") %>'>
      <br>
      Link(without &quot;http://&quot;)<br>
      <input name="Phone" type="text" value='<%=kayitlar("Phone") %>' size="50" maxlength="100">
      <br>
      Email<br>
      <input type="text" name="Email" value='<%=kayitlar("Email") %>' maxlength="50">
      <br>
      Comment <br>
      <textarea name="Comments" rows="5" cols="70"><%=kayitlar("Comments")%></textarea>
      <br>
      <label><span class="style35">
        <input name="Type" type="radio" value="1"
          <% if kayitlar("Type")=1 then response.Write("checked") %>
        >
      </span></label>
      <span class="style35"><br>
        <input type="radio" name="Type" value="2" <% if kayitlar("Type")=2 then response.Write("checked") %>
        >
      Site <br>
        <label>
          <input type="radio" name="Type" value="3"
            <% if kayitlar("Type")=3 then response.Write("checked") %>
          >
        </label>
      Program</label>
      <br>
      </span> <input type="hidden" name="sno" Value="<%=kayitlar("ContactID")%>">
      <input type="hidden" name="islem" value="DuzenleKaydet">
      <input type="Submit" Value="Update" onClick="alert('Process done');">
      </font><font size="2">
      <input type="Reset" Value="Reset">
      </font></span></td>
    </tr>
    <span class="style30">
  </table>
</span>
</form>
<p align=center><span class="style31"><font size="2"><a href="?islem=Ana">Menu</a>
  <%
bag.close
set bag = nothing
%>
  <% end if %>
<!-- HaberDüzenleAna -->
<% END SUB ' HaberDuzenleAna
'-----
%>

<% SUB DuzenleKaydet %>

```

```

<!-- Düzenle Kaydet -->
<%
first_name = Request.Form("FirstName")
last_name = Request.Form("LastName")
city_name = Request.Form("Phone")
email_name = Request.Form("Email")
comment_name = Request.Form("Comments")
t_name = Request.Form("Type")
sno = Request.Form("sno")
%>
<%
bag.open veritabani
sql="Select * from tblContacts Where ContactID= " & sno
ks.open sql, bag, 1, 3
ks("FirstName") = first_name
ks("LastName") = last_name
ks("Phone") = city_name
ks("Email") = email_name
ks("Comments") = comment_name
ks("Type") = t_name
ks.update
ks.close
set ks = nothing
bag.close
set bag = nothing
response.redirect("delete.asp?islem=Ana")
%>
<!-- Düzenle Kaydet -->
<% END SUB 'DuzenleKaydet %>
<%
'=====
%>
</font>
</span>
</body>
</html>

```

delete.asp içinde silme, güncelleme, ekleme işlemleri var. GuestBook.asp'de kayıt yapılmadan önce tüm alanların doldurulup doldurulmadığı kontrol ediliyor ve kullanıcının yorum gibi kısımlara HTML kodları girilmesi güvenlik amacı ile engelleniyor. Ekrana kayıt numaraları ile birlikte tüm kayıtlar geliyor. İsterseniz SQL* komutlarını kullanarak kayıt sıralama, kayıt silme, sayısal veriler girilip onların ortalaması veya minimum, maksimum gibi istatistiklerin tutulmasını, ayrıca kayıtlı kullanıcı oluşturup onların login veya logon ile sitenize girmeleri gibi hoş özellikler yapabilirsiniz. E-mail listesi tutma da buna benzer yapılabilir. Daha çok güvenlik gerektiren (kredi kartları bilgileri, e-ticaret gibi) alanlarda da SHTML – Secure HTML (Güvenli Yuva Katmanı veya SSL) kodlamasını öğrenmenizi tavsiye ederim.

JavaScript örneklerimizdeki tüm işlemleri *.asp olarak VBScript kodlama ile de yapabiliriz. "Ornekler\Diger \ASP\extra" dizininde Dr. Hakkı Öcal'ın çalışmalarını görebilirsiniz. Örneğin bilgisayarımızdaki sürücü listesini almak hemen hemen aynı:

```

<% Option Explicit %>
<HTML><HEAD><TITLE>ASP Listeleme Örneği</TITLE></HEAD><BODY>
<%Dim DosyaSistemi, Surucu, Suruculer
Set DosyaSistemi = CreateObject("Scripting.FileSystemObject")
Set Suruculer = DosyaSistemi.Drives
For Each Surucu In Suruculer%>
<b>Sürücü:</b> <%=Surucu.DriveLetter%><br>
<% If Surucu.IsReady = True Then%><b>Disk Adı:</b> <%=Surucu.VolumeName%><br>
<b>Boş alan:</b> <%=Surucu.FreeSpace%><br><% Else %>
<i>Sürücü hazır değil!</i><br>
<% End If
Next %>
</BODY> </HTML>

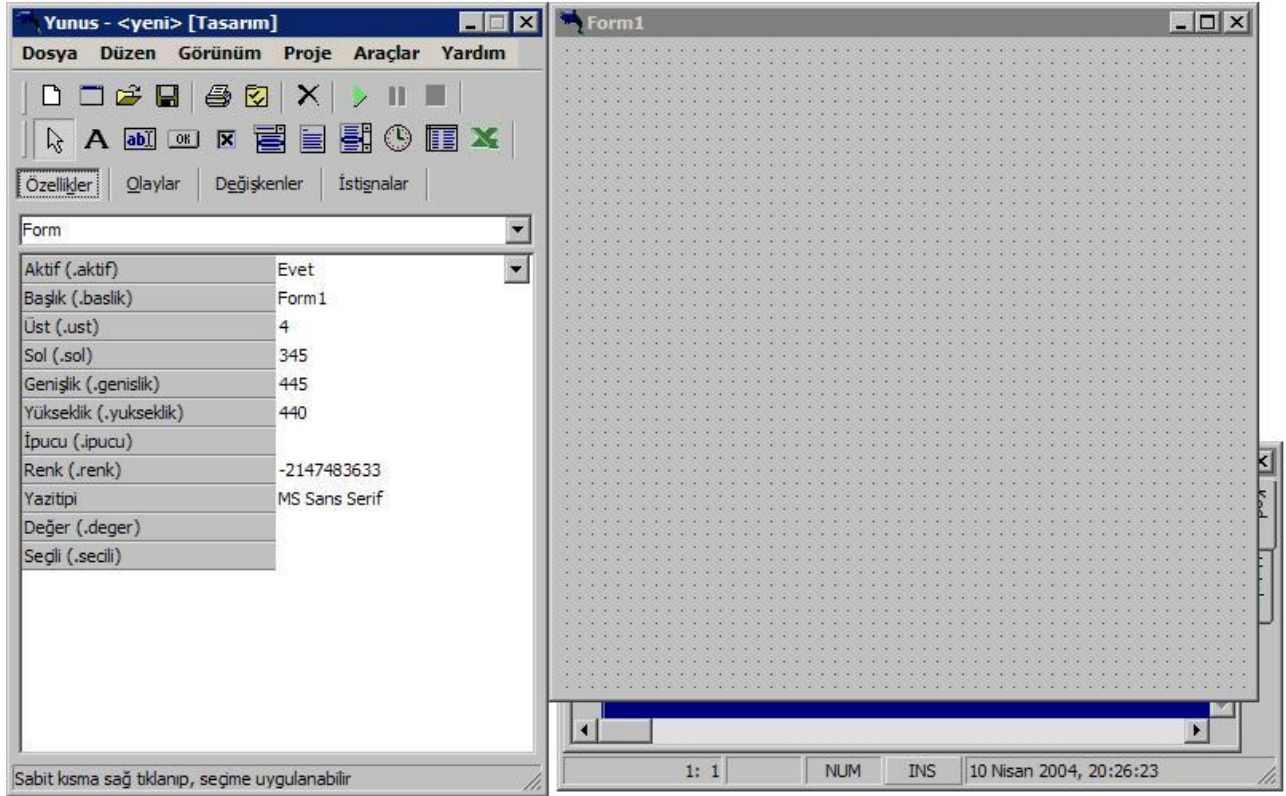
```

* Ek 2 – SQL kısmına bakınız.

13. Yunus Programlama Dili (Türkçe açık kaynak programlama projesi)

(Ornekler\Diger\Yunus\Yunus.exe)

Nihayet Nisan 2004 tarihinde kendi projemi de kitaba ekliyorum. Şu anki son sürümü 2.8 olan ve Delphi 6 ile oluşturduğum proje aslında bu kitabın bir sonucu da diyebiliriz. Aralık 2001'den bu yana, renkli kod desteği veren MemoX nesnesi dışında kendi emeğimiz ile birşeyler yapmaya çalışıyoruz. Bu bebek ile kanıtlamak istediğim, 1 MB boyut bile bulmayan küçük bir işlevsel programlama dili yapılabileceğidir. Multimedya ve veritabanı gibi güzel özellikleri daha kolay ve kullanışlı bir şekilde biraraya getiriyoruz. Fakat eksikleri daha var ve aslında bitmiş bir sürüm değil. Balinalar ile yarışabilmesi için sadece benim değil, bir çok insanın ilgilenmesine çabalıyorum. Özellikle birçok programlama sitesine üye olarak tanıtım yapıyorum. Hala tam istediğim gibi bir seviyede destek bulamadım.



Resim 17a – Yunus 2.8'in açılış görüntüsü

Yunus ismi Delphi'den geliyor. Delphi, dolphin yani yunus'a benzediğinden bu adı verdim. Dolphin diye bir müzik programım da var ve benimle ilgisi olmayan delphi-dolphin.com diye bir programlama sitesi de var. Özellikle delphiturk.com sitesinde yardım isteğim yanıtsız kaldı.

Programlama kod diline de SoleBee adını verdim. SoleBee'yi aslında modüler yapma düşüncem var. Yani istenen program içinde program yapabilme eklentisi yeteneği verilebilir.

Yunus'un daha geliştirilmiş hali olarak da BlueDream projem de var, fakat birçok nesnenin biraraya gelmesi ile yaptığım için, devamı çok zor gözüküyor. Çünkü Mart 2003 tarihinden itibaren başladığımdan bu yana Shareware Greateis Form Designer'ın daha yenisini yaptılar ve bu sürümünü eklemek için uğraştığımda uyumsuzluklarla karşılaştım. Yunus ise tamamen benim yazdığım kodlarla meydana geldiği için güncellemem daha kolay olabilir. BD içinde çok uğraştığım 3 boyut düzenleme penceresi ve nesne özellikleri pencereleri artık Yunus içinde de olabilir. Ayrıca yazdırma kısmı BD'de çok daha ileri seviyede.

Microsoft'un .NET teknolojisi hakkında o zamanlar (2001 yılı) hiçbir bilgim olmadığı halde benzer bir proje ile uğraşmışız. Tabii ki onların web entegrasyonu (*.aspx), PC yani kişisel bilgisayar dışında başka yerlerde de çalışabilme (cep telefonu, palmtop – avuçucu bilgisayarlar) özelliklerine gıpta etmekten başka yapabileceğimiz yok. Delphi.NET yani Delphi 9 sürümünde rahatlıkla Yunus 3.1'i açıp, çalıştırıyoruz. Sadece Qreport artık bulunmadığından sorun çıksa da SynEdit nesnesi bizi kurtarıyor. Yani hala kodları yazıcı ile çıktı alabiliyoruz.

Gelecek sürümlerde HTML, PHP veya ASP çıkışı ile internet, SQL desteği, OpenGL ile 3 boyutlu tasarım, daha gelişmiş veritabanı, daha fazla nesne ve daha fazla nesne özelliği de olacak. Yazdırma, kodlama, kod tamamlama da yeniden gözden geçirilecek.

Çoğunlukla Visual Basic benzeri kodlama sistemimiz var. Aslında hoşuma giden Pascal ve C++ komutları da ekledim. Sonuçta Türkçe veya başka bir dilde özelleştirilebilen basit bir *eğitim* amaçlı script dilimiz var. **Examples** klasörü içindeki örnekleri inceleyerek genel kullanım hakkında bilgi alabilirsiniz.

Birkaç ipucu:

Değişken türleri ve tanımlanması yoktur. Tüm değişkenler globaldir. Değişken adlarında sayı ile başlamak ve özel karakter kullanmak, boşluk bırakmak sorun çıkarmasa da tavsiye edilmez. Kullanıcı hatayı bilerek oluşturduğu kabul ediliyor.

Sayı (tamsayı ve gerçek), metin ve mantıksal (True - Evet, False - Hayır) kendiliğinden dönüştürülüyor. 1E3 rakamı 1000 anlamında kullanılabilir. Gerçek sayılarda "," virgül ile ondalık basamak ayrılabilir.

"=" ile atama ve karşılaştırma, satır sonlarında ";" yoktur. Aynı satıra ";" kullanarak daha çok komut yazmayı düşünüyorum.

"+", "-", "*", "/", "&" (metin birleştirme), "^" (exp - üs alma), "\" (div - tam bölüm), "%" (mod - kalan), "@" (mantıksal ve), "~" (mantıksal veya) hesaplama bağlacıdır. İççe hesap ve parantez kullanılabilir.

Değişken ve nesne özelliklerinden sonra "++" ve "--" ile artım eksiltme yapılabilir. ++ gibi, fakat şimdilik ++a yoktur. ++ deyişi a=a+1'den biraz hızlı işler.

Birleştirici hesaplamalar a+=2 (eşiti a=a+2) gibi tüm işlemler (+, -, *, /, ^, %, &, \, @, ~) için kullanılabilir. a+ =1 gibi operatör ve eşittir arasına boşluk bırakmayınız.

Hesap yönü soldan sağdır. Çarpma ve bölme önceliği yoktur. Bu gibi durumlarda parantez kullanılabilir. a=5+6*3-2 ==> a=21 //normalde; * (çarpma) önce yapılır, sonra da kalan yapılır. a=5+6*3-2 ==> a=31 //Yunus'da; 5+6, sonra 11*3, sonra 33-2... parantez ile a=5+(6*3)-2 asıl sonuç çıktığı görülebilir.

sin(gerçek), cos(gerçek), random(tamsayı), int(gerçek), not(mantıksal), len(metin), lower(metin), upper(metin), chr(tamsayı), asc(karakter), delete(metin, ilk, miktar), mid(metin, ilk, miktar), pos(metin, karakter), and(tamsayı, tamsayı), or(tamsayı, tamsayı), xor(tamsayı, tamsayı), shr(tamsayı, miktar), shl(tamsayı, miktar), time() ve date() fonksiyonlarını kod kısmında kullanabilirsiniz. Bu komutları özelleştirme penceresinde bulamazsınız.

Metinleri " işareti içine almak gerekebilir. Mantıksal değer olan "Evet" ve "Hayır" çift tırnak içine alınmalıdır.

Büyük küçük harf ayrımı yoktur. Komutlar, nesne özellikleri ve değişkenler istenilen büyüklükte karışık yazılabilir.

Eger komutu tek veya çok şartlı olarak kullanılabilir. Şart kısmında AND için "@" ve OR için de "~" gibi bağlaçları destekliyor.

Eger komutunda "=" ">" "<" "<>" "<=" ">=" kullanılabilir. İççe Eger (Degilse) kullanımı sorun çıkarmaz.

eEger şart, dogruSatir, yanlisSatir VB dilindeki iif gibidir. İç kısımlarda blokluk komutlar kullanılamaz.

Donder degiskenAdi=BaslangicDegeri, BitisDegeri, Artim şeklinde iççe kullanabileceğimiz döngü komutumuz var.

iken Şart şeklinde sonuç "Yanlış" olana dek dönen komut vardır. ikenBitti ile de bittiği belirtilir.

Kullanıcı kendi Fonksiyon'unu tanımlanabiliyor.

"/" açıklama satırı yapar. Satır içi açıklama satırı yapılabilir.

Etiket tanımlaması ile istenen satıra Git komutu ile dallanabilme imkanı vardır.

Deyimler arasına boşluk ve TAB yapılabilir. Fakat Git komutunun etiketi ":" ile bitişik olması gereklidir.

Yuzey komutu ile form üzerine metin, çizgi, nokta, dikdörtgen (kare), daire (elips), poligon yapılabilir. Fakat form üzerinden bir şeyler geçtiğinde silindiğini görebilirsiniz. Tazele komutu ile de formu temizleyebilirsiniz.

sesCal dosyaadi ile *.wav müzik dosyaları çalınabilir. *.mp3 çalışmıyor.

Tablo nesnesi ile *.xml (Extensible Markup Language) ve *.cbs (Client Data Set) dosyalarına erişip basit kayıtlar oluşturabilirsiniz. Şimdilik SQL (Structured Query Language) ve tablo tasarım penceresi (kısayolu: Ctrl+D) bitmedi.

ExcelTablosu nesnesi ile Excel dosyalarına ulaşabilir, istenen hücrenin değeri alınabilir, değiştirilebilir, sonrada kaydedilebilir. Yeni dosya oluşturma yoktur.

BenchmarkStart ve BenchmarkEnd kodları arası milisaniye olarak hız ölçümü verir. Sonuç Hata Penceresi'ne eklenir.

komut tekSatirKomut ile Yunus'un tek satırlık kendi komutlarını kullanabilirsiniz. Mesela; komut a=1

inport(adres) ve outport(adres, byteDeger) donanım olarak port kullanımı içindir. Örneğin 888 (378 onaltılık) adresi yazıcının veri portu olarak kullanılır. Başka portlar bilgisayarı kilitleyebilir.

Aşağıdaki listede bazı nesnelere atanabilen olay adları vardır. Form ilk açılış ayarları için "Yuklenince - OnLoad", kapanırken de "Kapanırken - OnClose" olayları oluşturulabilir.

Nesne	Yunus'daki olaylar	Delphi'deki olay eşdeğeri
Form	Yuklenince Kapanırken	OnShow OnClose

Etiket	Tiklanınca	OnClick
YaziKutusu	Tiklanınca	OnChange
Dugme	Tiklanınca	OnClick
OnayKutusu	Tiklanınca	OnClick
AcilirKutu	Tiklanınca	OnChange
YaziKutusu	Tiklanınca	OnChange
ListeKutusu	Tiklanınca	OnClick
Zamanlayici	Tiklanınca	OnTimer

Komutları değişken adı olarak kullanmamaya dikkat ediniz. Mesela Git, Donder, bitti özel kelimelerdir.
Proje adı olarak "yeni" kelimesi kullanmayınız.

Hedeflenenler:

- # Case komutu
- # Nesne oluşturma
- # Daha iyi nesne seçimi ve nesne boyutlandırma
- # Nesne ve değişken dizileri
- # Tam olarak çalışan birden fazla form kullanma
- # Daha geniş Yardım konuları
- # EXE ve HTML oluştururken tüm nesnelerin komutları ile birlikte çalışabilmeleri

Yunus.exe tek başına çalışsa da, aynı klasörde midas.dll (XML desteği için), zlportio.sys (Windows NT ve üstü için port kullanımı), yunus.chm (Yardım dosyası), *.exe oluşturma için gerekli Kalip klasörü olmalı, ve kod dökümü için bir yazıcı tanıtılmış olması daha iyi olur. Kopyalanan nesneler "~kopya.txt" adında kaydediliyor. Bu sayede bilgisayarı kapatsak da kopya nesneler saklı kalıyor. Ayarlar Windows kayıt defterinde değil, *.ini dosya yapısında "varsayılan.ayar" dosyasında saklanıyor. *.proje dosya uzantısı desteği ilk çalıştırmada oluşturulur, istenirse "Dosya*Seçenekler..."den silinebilir. Yani sisteme en az yük ve yer kaplaması ile çalışabiliyoruz. Her ne kadar 700KB olsa da hafızada 10MB gibi bir alan kaplıyor. BD ise 17MB kaplıyor.

Maalesef menü yabancı dil desteği için "menu.rc" dosyası yapmışız. Bu yüzden de isteyen kendi ülkesinin dil çevrimini yapmak isterse, Delphi'nin kurulu olması ve Yunus'un tekrar derlenmesi gerekiyor. Tasarım klasöründe derle.bat dosyası, brcc32.exe kullanarak menü değerleri *.res yani resource – kaynak haline getiriliyor.

ASPack denen başka bir shareware program ile de yunus.exe sıkıştırılıp üçte bir küçülmesi sağlanıyor. Bunu sadece Borland dosyaları için kullanabilirsiniz. Visual Basic ile yapılan exe dosyaları sıkıştırmıyor.

Yaklaşık 70 adet komutun "Proje*Komutlar" menüsünden özelleştirilmesi ise İngilizce ve Türkçe dilleri için yaptığım bir yenilik. Fakat özelleştirilince eski programlar istenen şekilde çalışmayabilir. Bir Fransız komutları kendi diline çevirse bile "varsayılan.ayar" dosyasını da diğer insanlara vermesi gerekiyor. Bu sebeple bayağı bir kargaşa çıkabilir. Eski ayarlara istediğimiz zaman geçsek de pek faydası olmayan ve kodların yavaşlamasına sebep olan bir seçenek. Hem de kopyalanan ve kesilen nesneler tam olarak sadece dili aynı olan projeye yapıştırılabilir.

Kod renklendirme ve yazıtipi özelleştirebilme de yapılabilir. Her projenin ayarı farklı olabilir ve "projeadi.ayar" adı ile saklanır. Burada da en çok karşılaşılan sorun, yazıtipinin başka bilgisayarda bulunmaması halinde kodların okunaklı olmaması oluyor. İlle de projeadi.proje açılmasına gerek yok, "Dosya*Aç"da projeadi.dfm de seçilse dosya adları aynı olduğu için proje açılır. Birden fazla form kaydını daha yapamadığım için bir sorun yok.

Çok karmaşık gibi gözüktü biliyorum, aslında diğer profesyonel uygulamalara göre çok basit sayılabilir. Derleme imkanı da olsaydı, script demekten vazgeçerdim. Internet Explorer nasıl metin tabanlı *.html dosyaları çalıştırıyorsa, Yunus da *.proje dosyalarını çalıştırıyor. *.exe oluşmadığı için minicik programlar yapıyoruz. Bu tür programlama dillerinin bir avantajı da sadece yorumlama programı güncellenerek projelerin çalışmaları daha iyileştirilebiliyor. Eğer daha hızlı çalışan algoritmalar yapabilirsek, uzun süren (döngüler gibi) işlemler daha çabuk bitirilebilir.

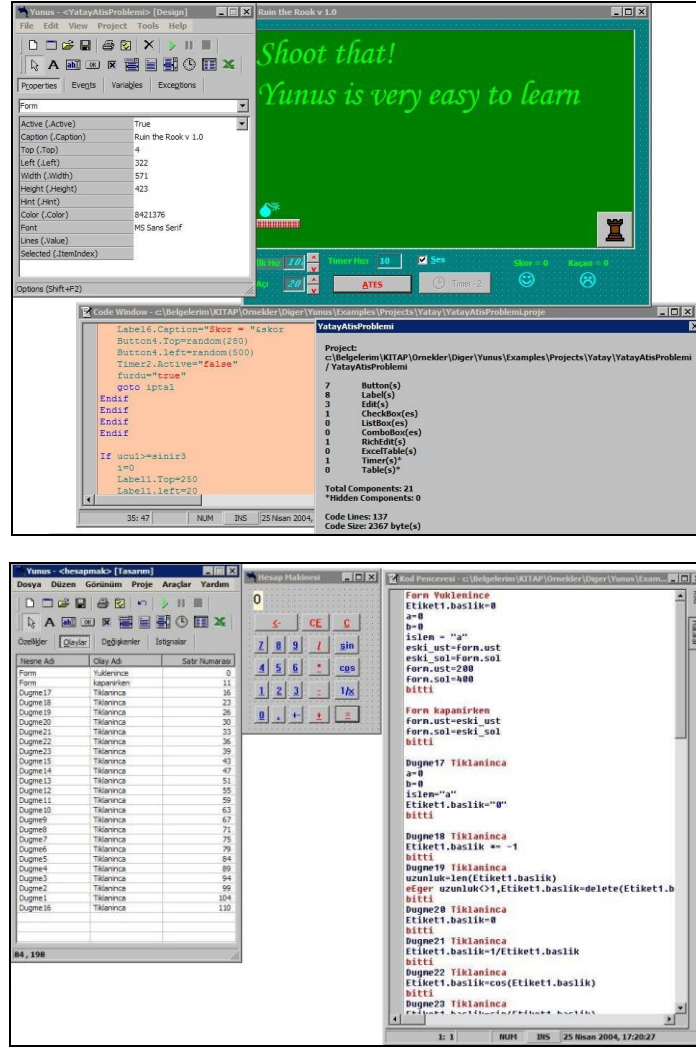
Tek tek menüleri ve komutları anlatmak zor, onun yerine örneklerden bazılarını anlatalım. Mesela YatayAtisProblemi.proje (İngilizce menü ve kod) ve hesapmak.proje (Türkçe menü ve kod).

Yatay atış fizik dersinden bildiğimiz bir problemdir. Buradan yola çıkarak basit bir oyun yaptık. Açık ve hız değiştirilerek hedef vurulmaya çalışılıyor. x ve y koordinatları şu formüllerle bulunuyor:

$$x = V_0 \cdot \cos(\alpha) \cdot t$$

$$y = Vo \cdot \sin(\alpha) \cdot t - \frac{1}{2} g \cdot t^2$$

$g = 10$ kabul edilirse $y = Vo \cdot \sin(\alpha) \cdot t - 5t^2$ de diyebiliriz.



Resim 17b – Yunus ile yapılmış iki örnek: YatayAtisProblemi ve hesapmak

Counter-Strike oyunundan aldığım ses dosyalarını belli anlarda çalıştırıyoruz. Eğer klasör doğru belirtilmezse sesler ve kale resmi olmadan yine program çalışıyor. Ya da şöyle diyelim, herhangi bir satırda bir hata oluştuğunda tüm programın kapanması yerine sadece o satır işlenmiyor, program çalışmaya devam ediyor. Hangi satırda hata olduğunu hata penceresinden (F6 kısayolu ile) bakabilirsiniz. Pek gelişmiş bir hata açıklaması yok. Hata penceresinde ayrıca Benchmark yani hız test sonuçları da görüntülenebiliyor. Kod bloğunun kaç milisaniyede çalıştığını görebiliriz. Ses çalmayla ilgili söyleyebileceğim diğer şeyler de aynı anda birden fazla ses çalınmadığı ve peşpeşe ses çalınmaya çalışıldığında da önceki sesin kapatıldığıdır. *.wav dışındaki diğer müzik ve ses dosyaları desteklenmiyor.

Bu örnekte program durdurulsa da "Persistent Variables – Kalıcı Değişkenler" seçeneği seçildiğinden, açılı ve hız bir sonraki çalıştırmada da kalıyor. Eğer program kaydedilirse daha sonraki Yunus çalıştırılmasında da aynı yerden devam edilebiliyor. Değişkenler penceresinde değişkenlere sağ tıklayıp sil, ekle, değiştir gibi seçeneklere ulaşabiliriz. Bu pencerede diğer dillerdeki "Watch – Gözetle" imkanı kendiliğinden hazır olduğu görülebilir. Arkaplanda neler olduğu daha belirgin. Aynı isimde değişken eklense de ilk bulunan aktif değişken olur. İçinde boşluk ve diğer bazı özel karakterler geçen değişkenler de kabul ediliyor.

Program hatası veya Yunus'un kendi hataları "Exceptions – İstisnalar" penceresinde tutuluyor. Sadece bilgilendirme amacı ile yapılmıştır. BD'de de bunları "error.log" diye bir dosyaya saklamayı tercih ettim. Ama ölümcül hatalarda (sabit diskin bozulması gibi) bu kayıtlar bile tutulmadan ana program kapanıyor.

Çift tırnaklar içine yazılanlar metin olarak algılansa da ilk başlarda içinde =, +, / gibi işlemler içeren kelimelerde onlar da çalıştırılmaya çalışıldığı için, zorlanmıştım. message "x=0" komutu maalesef ekrana x=0 mesajı çıkarmıyor. =

yerine : kullanabilirsiniz. Aşağıda görüldüğü gibi mesela "Kaçan ="&kacan gibi kullanım artık sorunsuz. Hatta "" kullanılmadan da atama yapılabilir. Ama o zaman da içinde =, + geçiyorsa hatalı bir kullanım olur. "True – Evet" ve "False – Hayır" çift tırnak içine alınmazsa değişken zannedilebilir.

Mantıksal işlemler ile bitisel işlemlerin and ve or'ları birbirine karıştırılabilir. If ve while komutlarının karşılaştırma deyimlerinde @ ve ~ kullanılırken, iki tamsayının bitisel olarak ve'lenmesi and(sayı1,sayı1) ile oluyor. Karşılaştırma gibi işlemlerde True ve False yerine 1 ve 0 kullanılıyor. Nesnelerin özelliklerini değiştirirken de True ve False kullanılıyor. Aslında ortak bir arabirim olabilirdi.

Sinüs ve kosinüs alınırken radyan değil doğrudan açı değeri kullanılıyor. Birçok dilde dereceden radyana çevrim yapmak zorunda kalıyorduk.

Yaptığımız projeyi Delphi'nin komut satırı yorumlayıcısı dcc32.exe ile *.exe yapmaya çalıştım. "Kalip" klasöründeki dosyaları temel alarak exe dosya oluşturuyor. Komutları tekrar SoleBee'den Pascal'a çevirme bitmese de, en azından form oluşturuyor. Başka bir dile çevirmek gereksiz gibi de görülebilir. Çünkü zaten isteyen istediği programı o dilde yapabilir. Aynı şekilde HTML çıktıda da komutlar JavaScript veya VBScript'e dönüştürülebilir, ama görünüm, neredeyse yüzde 95 gibi, asıl tasarladığımıza benziyor. İlk adımlarımız umut verici. Bunlara ek olarak da VB, VC ,ASP ve PHP dönüşümleri de olabilir. Dışardan Yunus'a exe ve html alma daha yok. Etrafta buna benzer bir şey hiç görmedim zaten. ResHacker gibi programlar formları filan okuyabilse de normal dilin kodlarına erişim hiçbir yerde yok. Disassembler (makine dilini gösteren) programları bu işi görüyor galiba. Bu işlerden önce nasıl exe dosya yapılıyor bunu keşfetmek gerekiyor.

Diğer programlama dillerinden farklı olarak "calistir komut" ile kendi SoleBee kodunu çalıştırabiliriz. Çok satırdan oluşan for, if, while, goto çalıştırılmaz. Eğer Yunus ile Yunus'a benzer bir programlama dili yapabilirsek bu son sürüm olacaktır herhalde.

Bir türlü yapamadığım diğer özellik de, daha fazla form ekleyebilme. Form oluşturma kolay olsa da, kod ekleme büyük problem. Kod ve formları kaydederken, isimleri ne olacak, her form için ayrı bir Yunus mu çalışacak, karara varamadım. Proje yöneticisi gibi bir ekleme silme penceresi olmalı. Ana form hangisi olacak, ilk açılışta kimler gözükecek...

İlk açılış ekranını basit tutmaya çalıştığım için, tek pencerede ana menü, araç çubukları, özellikler, olaylar birleştirilmiş halde. Ama yeni eklentiler oldukça nereye nasıl daha fazla sıkıştırırız, zor olacak gibi. Visual Studio .NET'in burada hoş bir özelliği var; alt pencereler otomatik gizlenebiliyor ve fare üzerlerine gelince ortaya çıkıyorlar. Eğer fazlaca kullanılan bir pencereniz varsa, Toolbox gibi, başlık çubuğundaki raptiyeye basarak sabitleyebilirsiniz. Macromedia FlashMX 2004'de de katlanan pencereler var fakat DotNET en kullanışlı arabirime sahip. İlk kez bu programları kullanan birisi bu kadar çok pencere arasında kafası karışabiliyor. Yunus ve BlueDream'da genellikle Delphi benzeri bir görünüm var. Yine de bu programları taklit amaçlı olmamaya gayret ettim.

Sık kullanılan bazı Delphi kodlarını da prosedür haline getirip, Yunus içinde komut olarak ekleyebiliriz. Örneğin internetten yani TCP/IP kullanarak başka bilgisayara bilgi yollama komutu ekleyebilirsiniz. netbaglan, netDosyaYolla, netMesajYolla gibi... Daha uçuk kaçık fikirler de olabilir. Mesela kablosuz erişim ile başka bir cihaz (telefon) ile veri alışverişi de olabilir. Yunus bu sebeple Delphi'nin makro programı da oluyor. Keşke kütüphane veya dosya dahil etme gibi bir imkanımız olsaydı. Başkalarının yaptığı güzel kodları kendi programımıza da ekleyebilirdik. Delphi'deki "Install ActiveX" seçeneği gibi, OCX dosyaları nesne paletimize de eklememiz profesyonel bir çalışma olabilirdi.

Yatay atış programının kodları:

Timer2 OnClick //bu proje ile ilk oyunumuzu yapıyoruz...

```
ilkhiz>Edit3.Caption
aci>Edit4.Caption
timer2.Value>Edit1.Caption
i+=0,35
x=int(ilkhiz*cos(aci)*i)
y=int(ilkhiz*sin(aci)*i-(5*i*i))
Label1.Left=x
Label1.Top=250-y
```

```
sinir1=Button4.Left+Button4.Width
sinir2=Button4.top+Button4.Height
sinir3=Memo1.Left+Memo1.Width
sinir4=Memo1.top+Memo1.Height
ucu1=Label1.Left+Label1.Width
```

```
ucu2=Label1.top+Label1.Height
furdu=""
```

```
If ucu1>=Button4.Left
If ucu2>=Button4.top
If Label1.left<=sinir1
If Label1.top<=sinir2
i=0
Label1.Top=250
Label1.left=20
Button4.Value=klasor&"ROOKW.BMP"
skor++
Label6.Caption="Skor = "&skor
Button4.Top=random(280)
Button4.left=random(500)
```

```

Timer2.Active="false"
furdu="true"
goto iptal
Endif
Endif
Endif
Endif

If ucu1>=sinir3
i=0
Label1.Top=250
Label1.Left=20
furdu="false"
kacan++
Label2.Caption="Kaçan = "&kacan
Timer2.Active="false"
goto iptal
Endif
If ucu2>=sinir4
i=0
Label1.Top=250
Label1.Left=20
furdu="false"
kacan++
Label2.Caption="Kaçan = "&kacan
Timer2.Active="false"
goto iptal
Endif
If label1.top<=Memo1.Top
i=0
Label1.Top=250
Label1.Left=20
furdu="false"
kacan++
Label2.Caption="Kaçan = "&kacan
Timer2.Active="false"
goto iptal
Endif
If label1.Left<=Memo1.Left
i=0
Label1.Top=250
Label1.Left=20
furdu="false"
kacan++
Label2.Caption="Kaçan = "&kacan
Timer2.Active="false"
goto iptal
Endif
iptal:
If Check1.Value="True"
goto SON_Sesli
else
goto SON_Sessiz
Endif

```

```

SON_Sesli:
If furdu="True"
playsound klasor&"yes.wav"
endif
If furdu="false"
playsound klasor&"no.wav"
endif
SON_Sessiz:
End

function Sifirla
i=0
ilkhiz=Edit3.Caption
aci=Edit4.Caption
Timer2.Active="false"
Label1.Top=250
Label1.Left=20
klasor= "C:\Yunus\Examples\Projects\Yatay\"
Button4.Value=klasor&"ROOKB.BMP"
end

Form OnLoad
sifirla
skor=0
kacan = 0
Label6.Caption="Skor = 0"
Label2.Caption="Kaçan = 0"
Button4.Top=267
Button4.Left=491
If Check1.Value="True", playsound klasor&"enter.wav", ""
End

Button1 OnClick
sifirla
If Check1.Value="True", playsound klasor&"shoot.wav", ""
Timer2.Active="true"
End

Button3 OnClick
edit3.Caption += 1
End

Button6 OnClick
edit3.Caption -= 1
End

Button8 OnClick
edit4.Caption += 1
End

Button7 OnClick
edit4.Caption -= 1
End

```

Hesap Makinesi programının kodları:

```

Form Yuklenince
Etiket1.baslik=0
a=0
b=0
islem = "a"
eski_ust=form.ust
eski_sol=Form.sol
form.ust=200
form.sol=400

```

```

bitti

Form kapanirken
form.ust=eski_ust
form.sol=eski_sol
bitti

Dugme17 Tiklanınca
a=0

```

```

b=0
islem="a"
Etiket1.baslik="0"
bitti

Dugme18 Tiklaninca
Etiket1.baslik *= -1
bitti
Dugme19 Tiklaninca
uzunluk=len(Etiket1.baslik)
eEger
uzunluk<>1,Etiket1.baslik=delete(Etiket1.baslik,uzunluk,1),Etiket1.baslik=0
bitti
Dugme20 Tiklaninca
Etiket1.baslik=0
bitti
Dugme21 Tiklaninca
Etiket1.baslik=1/Etiket1.baslik
bitti
Dugme22 Tiklaninca
Etiket1.baslik=cos(Etiket1.baslik)
bitti
Dugme23 Tiklaninca
Etiket1.baslik=sin(Etiket1.baslik)
bitti

Dugme15 Tiklaninca
eeger Etiket1.baslik=0,Etiket1.baslik="",
Etiket1.baslik&="1"
bitti
Dugme14 Tiklaninca
eeger Etiket1.baslik=0,Etiket1.baslik="",
Etiket1.baslik&="2"
bitti
Dugme13 Tiklaninca
eeger Etiket1.baslik=0,Etiket1.baslik="",
Etiket1.baslik&="3"
bitti
Dugme12 Tiklaninca
eeger Etiket1.baslik=0,Etiket1.baslik="",
Etiket1.baslik&="4"
bitti
Dugme11 Tiklaninca
eeger Etiket1.baslik=0,Etiket1.baslik="",
Etiket1.baslik&="5"
bitti
Dugme10 Tiklaninca
eeger Etiket1.baslik=0,Etiket1.baslik="",
Etiket1.baslik&="6"
bitti

Dugme9 Tiklaninca
eeger Etiket1.baslik=0,Etiket1.baslik="",
Etiket1.baslik&="7"
bitti
Dugme8 Tiklaninca
eeger Etiket1.baslik=0,Etiket1.baslik="",
Etiket1.baslik&="8"
bitti
Dugme7 Tiklaninca
eeger Etiket1.baslik=0,Etiket1.baslik="",
Etiket1.baslik&="9"
bitti
Dugme6 Tiklaninca
eeger Etiket1.baslik=0,Etiket1.baslik="",
Etiket1.baslik&="0"
bitti

Dugme5 Tiklaninca
ara=int(Etiket1.baslik)
eEger ara=Etiket1.baslik,Etiket1.baslik&="," , ""
bitti

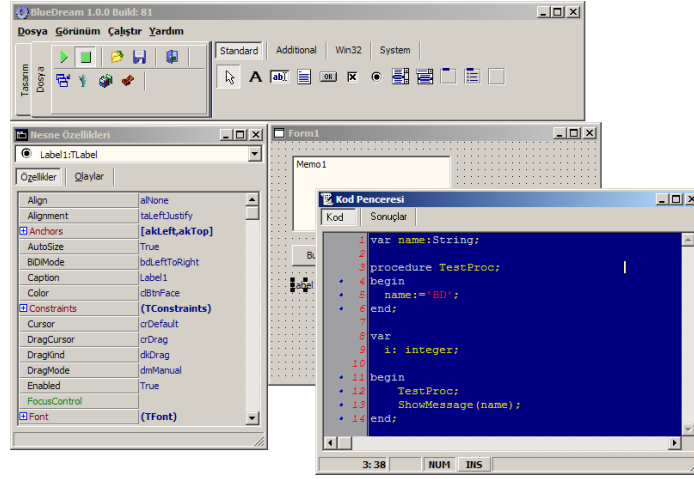
Dugme4 Tiklaninca
a=Etiket1.baslik
islem="a"
Etiket1.baslik="0"
bitti
Dugme3 Tiklaninca
a=Etiket1.baslik
islem="c"
Etiket1.baslik="0"
bitti
Dugme2 Tiklaninca
a=Etiket1.baslik
islem="m"
Etiket1.baslik="0"
bitti
Dugme1 Tiklaninca
a=Etiket1.baslik
islem="b"
Etiket1.baslik="0"
bitti

Dugme16 Tiklaninca
b=Etiket1.baslik
eeger islem="a",Etiket1.baslik=a+b, ""
eeger islem="c",Etiket1.baslik=a-b, ""
eeger islem="m",Etiket1.baslik=a*b, ""
eeger islem="b",Etiket1.baslik=a/b, ""
bitti

```

Her iki örnek 100 satırı (yaklaşık 2000 Byte) aşan kod ile yapılmış. Bu programları Delphi ile yapsaydık, yine yaklaşık bu kadar veya daha fazla yazarak elde edebilirdik. Önemli olan daha doğrusu asıl hedefimiz, diğer dillere üstünlük sağlama değil, işlerimizi basit yollardan halletmeye çalışmak. İlk kez görsel programlama yapan biri için Yunus rahat ve basit bir programlama ortamı.

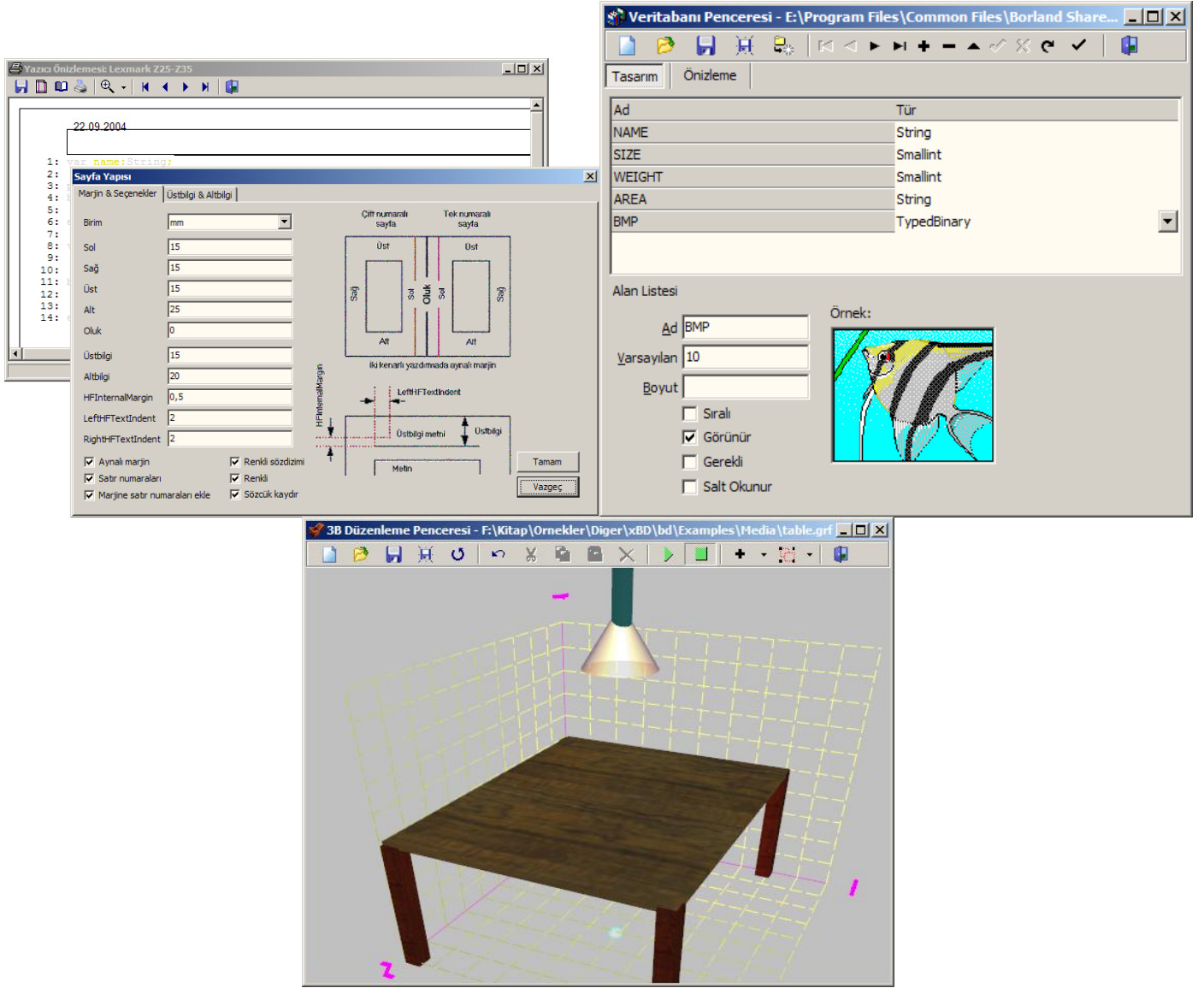
BlueDream projem ile ilgili birkaç resim de şunlardır:



BD ile Yunus kadar çok program yapma imkanı yok. Çünkü SoleBee'yi burada kullanmadım. DelphiWebScript II 1.1 nesnesi ile neredeyse Delphi ile aynı kod ortamı oluşturdum. Ama form tasarımı ile kod birlikte çalışmıyor. Form tasarımı da Greatis Form Designer 3.5 nesnesi ile yapıldı. Kod renklendirilmesi SynEdit1.1 nesnesi ile. Language v1.0 nesnesi ile de İngilizce ve Türkçe dillerine destek sağladım. 3 boyut tasarım softGL'den örnek alındı. Yunus da ise sadece MemoX ile diğer insanların yaptığı bir nesneyi kullanırken, BD projesinde internetten yaptığım uzun araştırmalar sonunda bulduğum nesne ve kodları kullandım. Pencerelemin hemen hemen hepsinde eksikler ve hiç bitmemiş kısımlar var. Ama üzerinde çok uğraştığım kısımları sıralamak gerekirse; "Nesne Özellikleri", "3B Düzenleme Penceresi", "Yazıcı Önizleme" diyebilirim. Genel görünüm ve menülerin tasarımı da az da olsa zamanımı aldı.

Bu projenin devam etmeme nedenlerinin başında yeni çıkan ve güncellenen nesneleri BD içine dahil etmede çok zorlanmam ve piyasada "CD2\Programlama\ScriptDiller\delphiscrptfull.exe" gibi çok daha iyi hazırlanmış ve çalışan örnekler olması. Ama ileri bir zamanda yeni kazandığım deneyimleri Yunus'a aktararak projeyi devam da edebilirim.

Yunus ile PHP ve ASP gibi dillerde kullanılan basit form öğeleri gibi çıktı alınabildiği için, BD'deki gibi gelişmiş "Nesne Özellikleri"ne gerek yok da denilebilir. Ya da basit özellikler ve gelişmiş özellikler diye programcıya seçenek bırakılabilir. BD'deki "Nesne Özellikleri"nde ValueListEditor nesnesinin üzerinde çok uğraştım. Tek tek çoğu nesnelere ait alt özellikleri bulup, düzenlenmelerine çalışıyoruz. Her ne kadar tüm özellikler ve olaylar listelense bile form üzerindeki nesneler ile iletişim tam değil. Mesela Form'un Action özelliği aktif değil.



Resim 17c – BlueDream projesi

3 boyut yani OpenGL rahatlıkla GeForce ve ATI ekran kartlarında çalışıyor. Küre, dikdörtgen prizma, yay, yazı, üçgen, tüp, simit, disk ve ışık gibi temel nesnelerimiz var. Projemin bitimine yakın çok doku ekleyebilme ve nesneleri "Nesne Özellikleri" penceresi ile birleştirmeye çalışıyordum. Formlar nasıl sonuçta çalıştırılıyorlarsa, bu penceredekileri denemek için sahne içinde dolaşma, "Çalıştır" kısmını yapmaya çalışsam da bitmedi.

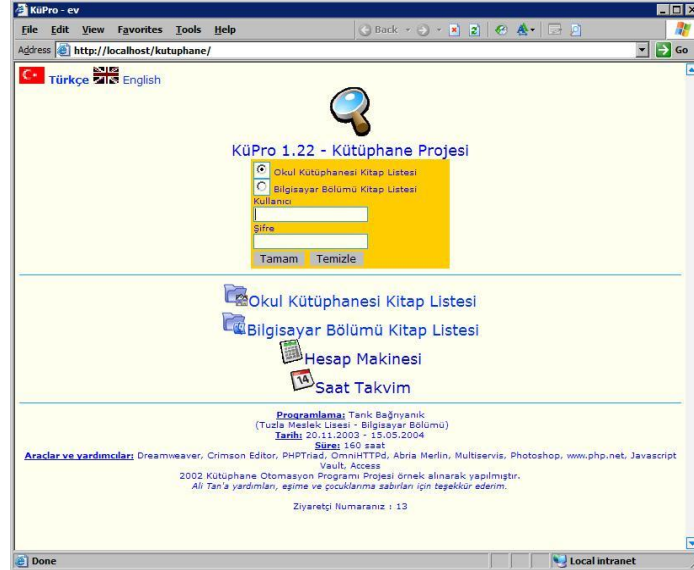
En son bitirmek istediğim, nesnelere animasyon imkanı ve onlara ait kod yazılabilme özelliği kazandırmak. Yani zamanla kendimiz de DarkBasic (DirectX kullanıyor) gibi oyun hazırlama platformu veya benzeri işlemleri yapabileceğimiz rahat bir ortamda program yapabiliriz. Tek başıma çok gelişmiş şeyler yapamam. Ne kadar uğraşsak da hep ilkel kalacak.

Geliştirildiğim kısımları söküp başka programlarda kullanmayı maalesef zor buluyorum. Programlama kısmında modülerlik kuralına pek uymadığımız için. Hayatı zorlaştıran hep kendimiz oluyoruz. Basit ve kolay işler yaparak, diğer programcılara örnek olmak gerekir. Benim örnek aldığım kaynaklarda genellikle açıklamalar ve düzenli programlama olduğundan az bir uğraşla kendi programıma bunları ekledim.

14. KüPro (Web tabanlı uygulama geliştirme)

(Ornekler\Diger\kutuphane192.zip)

PHP (Personal Homepage) ile yaptığım bu program ile bir kütüphane için gerekli kitap kaydı, kitap ödünç verme alma işlemlerinin takibi ve istatistik çalışmaları yapılabilir. PCNet dergisinden bulduğum birçok Mac ikonları süslü püslü olmasını sağladı. PHP tek başına bu kadar görsellik sağlamıyor. Ayrıca birçok JavaScript ve DHTML kodu da arkaplanda çalışıyor. Bunlar için JavaScript Vault (myplace.50g.com) kullandım. Toplam 5200 satır üzeri kodu yazarken, Notepad yerine de Crimson Editor'ü kodları renklendirdiği ve daha pratik olduğu için kullandım.



Resim 18a – KüPro giriş ekranı

ASP ile dinamik sayfa yapmak çok kolay, aynı şekilde PHP'de *.mdb dosyaları ODBC ile veritabanı kullanılarak işlerimizi görebiliriz. PHP tek başına bir anlam ifade etmiyor. Ana makine programı yine bedava olan Apache, veritabanı olarak da MySQL kullanılıyor. Genellikle de PHP Linux veya Unix üzerinde çalışıyor. Yani bu dil hem açık kaynak kodlu hem de bedava bir tasarım dili. Windows üzerine de bu programlar kurulabiliyor. Fakat Linux kadar performans alınamayabilir.

Toplu paket adında AbriaMerlin, PHPTriad ve multiservis (Türkçe) başlangıç ayarlarını bilmeyenler için hazır paketlerdir. Toplam 40-50MB arasında olan bu programlarda bol miktarda örnek ve öğretici doküman var. Az yer kaplayan birçok bedava alternatif de var. Mesela Abyss Web Server ve BadBlue gibi. OmniHTTP ile PHP de kendiliğinden kuruluyor ve kurulumdan kısa süre sonra da "<http://localhost>" da test sayfalarına bakabiliyorsunuz. omnicron.ca sitesini ziyaret ediniz.

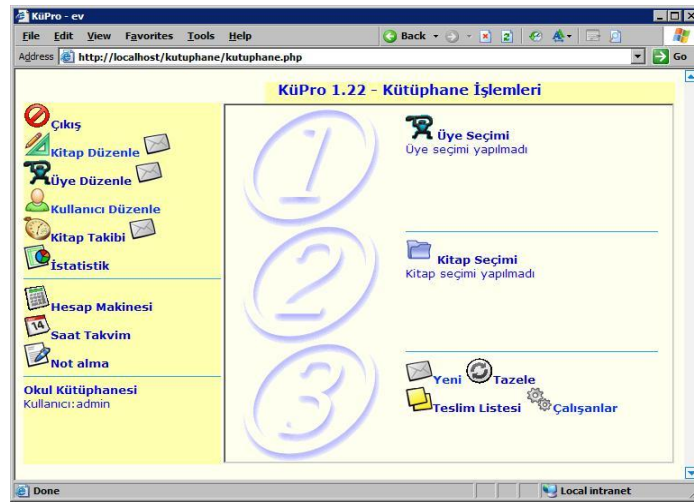
Eğer hem ASP hem de PHP aynı anda çalışsın diyorsanız IIS ve PWS için ayarlar internet sayfalarından öğrenilebilir. Bu kadar imkan varken önemli olan hız ve az yer kaplama ise en iyisi OmniHTTP diyeceğim. Bu arada Linux üzerinden de ODBC (iODBC ile) ve ASP çalıştırılabilir. İnternette arama yapınca birçok seçenek karşınıza geliyor.

Diğer bir problem de Netscape, Mozilla, Opera gibi browser – tarayıcılarda iframe ve stiller gibi konular tam desteklenmiyor. Bu sebeple Unix kullananların ekranlarında siteniz berbat görünebilir.

KüPro 1.22 - Bilgisayar Bölümü

Sıra D.No	Kitap Adı	Sayı	Yer	Tür	Yazar	Yayınevi
1)	100 Transistör Kataloğu	1	K1	Katalog		TOSHIBA (JAPONICA)
2)	101 Entegre Kataloğu	1	k1	Katalog		DATEL (JAPONICA)
3)	102 Entegre Kataloğu	1	K1	Katalog		TOSHIBA (JAPONICA)
4)	103 Fet kataloğu	1	K1	Katalog		TOSHIBA (JAPONICA)
5)	104 C-MOSIC Entegre Kataloğu	1	K1	Katalog		TOSHIBA (JAPONICA)
6)	105 Opamp Kataloğu	1	K1	Katalog		DATEL (JAPONICA)
7)	106 C-MOSIC Entegre Kataloğu	1	K1	Katalog		TOSHIBA
8)	107 Transistör Kataloğu	1	K1	Katalog		SGS - ATES GROUP
9)	108 Transistör Kataloğu	1	K1	Katalog		TOSHIBA
10)	109 Yarı iletken Kataloğu	1	K1	Katalog		MOTOROLA
11)	110 Analog Bileşik Devreler Kataloğu	1	K1	Katalog		PMI
12)	111 Birleşik Devreler Kataloğu	1	K1	Katalog		MOTOROLA
13)	112 Pascal programlama kitabı	1	P1	Pascal Programlama	Nell Dale	HEATH

Resim 18b – Kitap listesi



Resim 18c – Kitap takibi

KüPro 1.22 - Kütüphane İşlemleri

Kayıt - 24/482

Demirbaş No	123	Yer	P1
Yazar		Sayı	1
Kitap Adı	Turbo Pascal/ User s	Cilt	
Yayınevi	Borland	Sıra	
Tür	Pascal Programlama	Fiyat	
Düşünce		Düzen kitap ekile	

Lütfen kırmızı alanları boş bırakmayınız

Resim 18d – Kitap düzenleme ekranı

Sonraki versiyonlarda daha hızlı ve veritabanları az yer kaplayan MySQL kullanabiliriz. Neyseki şimdilik sadece birkaç bin kayıt ile uğraşıyoruz. Milyonlarca kayıt olsaydı, MySQL ancak işlemleri becerebilirdi.

Windows'da projenin çalışması için şu sırayı takip ediniz:

- 1) kutuphane.zip dosyasını "apache\htdocs" klasörüne "kutuphane" isminde klasör oluşturarak açınız.

- 2) Denetim Masasındaki ODBC simgesini kullanarak yeni bir Sistem DSN (Datasource Name – Veri kaynağı adı) ekleyelim. İsmine "veri" diyerek "kutuphane\inc\kutuphane.mdb" dosyasına yönlendirip kapatalım.
- 3) Windows dizinindeki php.ini dosyasındaki "register_globals = on" satırını olmasına dikkat edelim.
- 4) Projeyi "<http://localhost/kutuphane>" diye açtığımızda, oturum kayıtlarını kök dizininde sakladığımız için muhtemelen "c:\\" de sess_ef2b2b6ded... gibi dosyalar oluşacak. Daha sonra bunları rahatlıkla silebilirsiniz. Linux'da ise kök dizine yazma hakkı verilmez, o sebeple oturumlar "/tmp" gibi yerlere yönlendirilmelidir. Her *.php dosya içindeki `session_save_path("/tmp");` olarak düzenlenebilir.
- 5) nobetçi ismi ve şifresi ile giriş yapılırken kitap alış verişi yapılabilir; admin ise kullanıcı açabilir, kitap ekleyebilir, silebilir, kitap alış verişi yapabilir, öğrencileri üye yapabilir; kullanıcılar ise kitap ekleme, silme ve kitap alış verişi yapabilir, öğrencileri üye yapabilir.
- 6) "apache\htdocs\kutuphane\inc" klasöründeki `turkish.php` ve `english.php` dosyalarında gerekli düzenlemeler yapılarak başka dillere destek veriliyor. Hem de başka bölüm var ise, mesela matbaa bölümünün kütüphanesinde kullanılacaksa, `turkish.php` içindeki Bilgisayar geçen yerleri Matbaa diye değiştirilmek yetiyor.
- 7) Microsoft Access ile "apache\htdocs\kutuphane\inc" klasöründeki `kutuphane.mdb` düzenlenebilir. Yeni kütüphane açanlar buradan eski kitapları, kayıtları silebilirler.
- 8) Bu projede kullanıcı girişi için kullanıcı:admin, şifre:123 kullanabiliriz, `kutuphane.mdb`'nin şifresi de `my_library`. Şifreleri `md5` komutu ile kriptolu işlemine sokabiliriz veya kendi algoritmamız ile de session – oturuma kaydedebiliriz.

SQL desteği sayesinde istatistik kısmında güzel sonuçlar elde ediyoruz. Mesela;

```
$sql= "SELECT UyeNo,count(UyeNo) as toplam FROM teslim where Demirbasno='kitapsiz' and UyeNo<>"
group by UyeNo order by count(UyeNo) desc";
```

En çok çalışmaya gelen üyelerin numaralarını veriyor. İsimlerini de başka bir tablodan alıyoruz:

```
$sql2="select uyeno, adi, soyadi from uyeler where uyeno=$uyeno2 ";
```

Genellikle en çok form elemanlarına ulaşma, içlerini güncelleme, uyarılar, metin kutuları, onay kutularının çalışmaları gibi konularda JavaScript'e mecbur kalmak tuhaf. Görsel kısımlarda PHP'nin hiçbir yeteneği yok. Extensions – genişleme eklentilerini de çalıştıramadığım için grafik komutları gibi güzel unsurlardan yoksun kaldım. Ayrıca `php.ini` ve `httpd.conf` gibi dosyaların içine girip, ayar yapmak bana zor geliyor. PHPConfig (analogx.com) biraz yardımcı olsa da işin içinden çıkamadım. Unix kullananlar en basit ayarı bile not defteri benzeri programlar ile yapmaya alıştırlar, ama Windows kullananlar için anlaşılması zor.

```
if ($baski!="evet") echo "&nbsp;<a href=# onClick=history.go(-1)>Geri</a>";
```

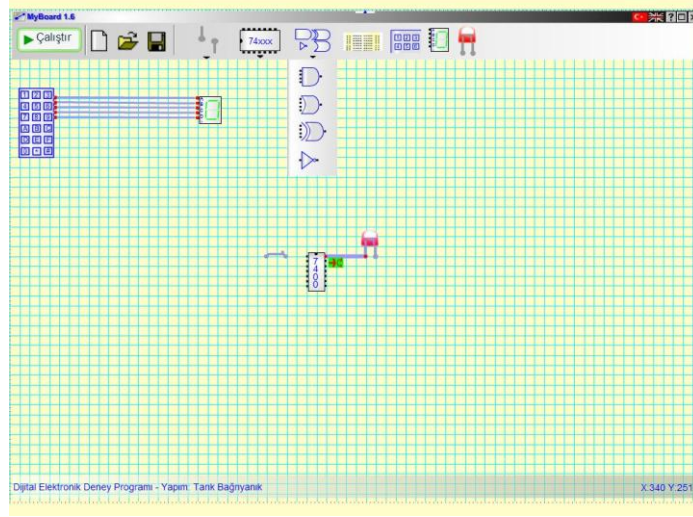
Buradaki gibi HTML, JavaScript ve PHP bir arada kullandım.

Türkçe ve İngilizce desteğini dış dosyaları "include" ile ederek sağlarken, aslında bu programı başka amaçlar için de kullanma imkanı ortaya çıkarılıyor. Mesela bilgisayar kelimelerini elektronik ile değiştirdiğinizde, elektronik bölümünün kütüphanesi de yapılmış oluyor. Tabi tabloları da ona göre isimlendirilmeli. Ayrıca Soru Bankası, Online Test Programı gibi projelerde de aynı altyapıyı kullanıyoruz. Programımız yerel de çalıştırılabilir, intranet, internet üzerinden de kullanılabilir.

15. MyBoard (Flash ile dijital devre tasarımı)

(Ornekler\Diger\mb\mb.zip)

21 Ocak 2004 ile 30 Mart 2004 tarihleri arasında Macromedia FlashMXProfessional 2004 ile yaptığım bu örnekte amaç basit devreler tasarlayıp onları çalıştırabilmek. Çalıştırma ve kaydetme maalesef bitmedi. Pek bitecek gibi de gözüküyor. Flash arabirimine bir türlü alışamadım. Minicik pencere ve yazılar, bir sürü kısayol, kısıtlı yardım dosyaları ve sonuçta gittikçe yavaşlayan bir *.fla dosya beni bıktırdı. Biraz daha kafa yorsaydım ve sabretseydim, "Çalıştır" da olabilecekti.



Resim 19a – MyBoard 1.6 tasarım penceresi

Kısayolları:

Boşluk	= Çalıştır / Dur
Ctrl + F	= Tam Ekran
Ctrl + W	= Çıkış
Shift + sürükley	= Elemanı serbest taşı
Oklar	= Elemanı 10 pixel taşı
Shift + Oklar	= Elemanı 1 pixel taşı
Delete	= Seçili elemanı siler
Esc	= Pencere görünümü

Bu programı da bedava ve açık kaynak kodlu proje olarak yaptım. Menü ve ipuçlarında, İngilizce ve Türkçe dili desteği var. english.txt dosyasını değiştirerek başka dilde de destek sağlanabilir. Türkçe dosyasını not defteri ile işlerken UTF-8 kodlama seçilerek kaydetmeyi unutmayın. Yoksa Türkçe karakterler Flash'da okunmuyor.

Yerleştirilen nesnelerin iğne veya bacak uçları tıklanınca kablo durumuna geçilir. Tekrar aynı uç tıklanınca vazgeçilir, ya da başka bir nesneye bağlanır. Kablo durumunda sürüklenme imkanı iptal edilir. Köşeli kablo veya kabloya başka bir şey bağlama, çok telli bağlantı yok. ISIS Proteus kullananlar burada hayal kırıklığına uğrayabilir.

Nesnelere sağ tıklanıp, Cut, Copy, Paste ve Delete gibi işlemleri yapabiliriz. Bu işi "Clipboard.as" dosyası hallediyor. Türkçe menü yapamadım. Paste yani yapıştırmak için fare ile başka bir boş alan sağ tıklanır ve komut verilir. Yeni nesnenin adı biraz tuhaf rakamla bitiyor. Bu isimlendirmede de çok uğraşmadım.

Led elemanının içinde direnç olduğunu varsayıyoruz. Yani normalde direnç ile kullanılmalıdır. Led doğrudan kaynağa bağlanınca üzerinden fazla akım geçer ve ömrü kısalmır. 7 parçalı Led gösterge de (Display) 7447 gibi elemanlarla kullanılır. Yine programımızda içinde varolduğunu düşünerek tasarım yapıyoruz. Zaten bu arada tüm nesneler de gerçek boyutlarında değiller. Baskı devre hazırlama programı amacımız yoktu. Programımıza "Devre Test Programı" da diyebiliriz.

XML dosyalara erişim neredeyse çok basit iken yazma işin içine girince PHP, JavaScript gibi yardımcı dillere ihtiyaç duyuluyor. Güvenlik neden ile olsa gerek. Ama stand-alone dosya yani exe dosya olarak bilgisayarımızın diskinden kullanacağımız program için başka çözüm bulmak gerekli. PHP ile de site hazırlarken HTML ve JavaScript olmadan çoğu şeyleri yapamıyoruz.

```
XML_var = new XML ();
XML_var.load ("empty.xml");
XML_var.onLoad = displayXML;
function displayXML () {...}
```

Bu projede Flash'ı tercih etmemin en büyük sebebi, pencereye sağ tıklayıp büyüt, küçült komutlarını verebilmemiz. Kalite bozulmadan bu işlemler yapabiliriz. Normalde programlama dillerinde grafik olarak scale (ölçeklendirme) yapmanın bir sürü tuhafıkları oluyor. Flash içindeki tüm çizilen nesneler(yazılar, çizgiler, daireler, poligonlar... - bitmap resimler ise net olmayabilir) vektörel olarak yapılır. Ses ve dışarıdan film alma gelişmiş. Fakat burada hiç kullanmadım.

En çok kafamı yoran elektrik akımının nasıl kablolar üzerinde akacağı oldu. Birden fazla Vcc (+5V, 1, yada artı) ve Gnd (0V, 0, Toprak yada eksi) olunca ne olacak? Eğer açık uç varsayılan olarak 1 kabul edilirse dijital devre elemanlarının davranışları ona göre değişmeli. Ya kısıadevre durumu nasıl değerlendirilecek? Nesnelere mi davranış vermeliydim yoksa bir prosedür yapıp devamlı tarama mı yapsaydım karar veremedim. Birbirine değen ve bağlı elemanların bir listesini "Çalıştır" deyince çıkarıyoruz. Bu metin dizisi halindeki listeyi işlemek zor geliyor.

ActionScript 2.0 ile ilgili de pek iyi anılarım olmadığını söyleyebilirim. Ama en sevdiğim şey Ctrl+Shift+F kısayolu ile "Auto Format" kod yazımını derleyip toparlaması oldu. Hem yazım hatası kontrolü yapıyor hem de kod okunaklılığını artırıyor. Tabi iç içe çok blok varsa daha standart görünüm elde ediyoruz. Biçimlendirmede bazı seçenekler "Auto Format Options..." ile görülebilir. Yine de kod yazımı, VB veya Delphi'nin yanında çok sönük ve yardımseverlikten uzak. Nereye ne yazdığımı ilk başlarda bir türlü anlayamadım. Movie Explorer(Alt+F3) ve Library (Ctrl+L) karışık, kafa karıştırıcı... macromedia.com'dan tüm pdf dosyaları indirsem de bunları pek kullanmadım. google.com'dan arama ile bir sürü pratik bilgiye ulaşabilirsiniz. Flash'ın yetenekleri bol olsa da, neredeyse yarısını kullanmadım. Kütüphanesindeki bir sürü güzel elemanlar pek işimi görmedi.

Resim 19b'deki gibi 17" ekran 1024*768 çözünürlük yetersiz... F4 ile tüm pencereler gizlense de neredeyse hiç bu durumda kalamadım. Her şeye kısayol tuşları verilmesine rağmen genellikle faremizi tıkır tıkır çalıştırmak zorunda kalıyoruz. Flash ile animasyon filmi yapanların işi daha zor. Bu projede sadece 2 katman var ve tek frame - çerçeve ile çalışıyoruz. Programların çoğu MovieClip - film kliplerine yazıldı. Belki de projemizi yavaşlatan şey, klip içinde klipler yapıp her birinde de program çalışması olabilir. Özellikle 14 bacaklı entegrelerden bir iki tanesini kullanınca neredeyse kilitleniyor.

Nesneleri eğer Transform - dönüştürme işlemine sokmadıysanız, ana çerçeve ile aynı x y birimine sahip oluyorlar. Fakat yükseklik ve genişlikle oynayınca nesnenin uzayı, birimi farklı oluyor. Bir de bunların eşdeğerini bulmaya çalışırken iyice işin içinden çıkamadım. Sonradan nesne oluşturduğumuzdan istediğimiz yere getirmek, kablo çekmeye çalışmak zor oldu.

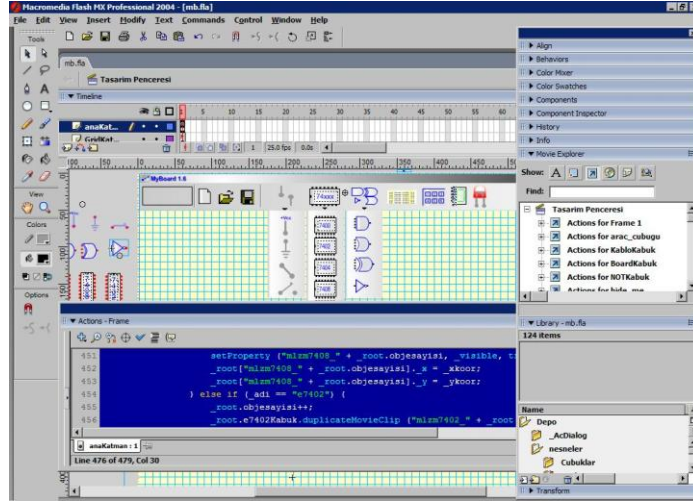
anaKatman ismindeki katmanımızda tanıladığımız değişkenler ve fonksiyonlar global oluyorlar.

```
var ilkNoktaX:Number = 0;
var ilkNoktaY:Number = 0;
var sonNoktaX:Number = 0;
var sonNoktaY:Number = 0;
```

C dilindeki gibi büyük küçük harf duyarlılığı, değişken isimlerinde ve fonksiyon adlarında var. Pascal mı desem Basic mi desem, karışık işte... Bir iki örnek yaptıktan sonra eliniz alışıyor.

Fare ve klavye olaylarının isimlendirilmesi garip, Delphi'deki onKeyDown, onMouseMove daha anlaşılır:

```
on (press) { //fare tuş basımı, bu olay yazılan kliplerde fare imleci el olur
}
onClipEvent (keyDown) { //klavye tuşu basımı
}
```



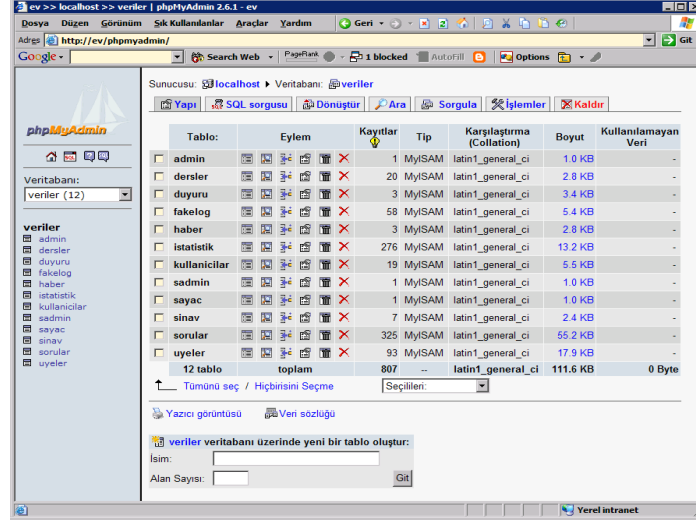
Resim 19b – MyBoard 1.6’un tasarlandığı pencere!?

ResourceHacker gibi programlar ile exe dosyamız üzerinde değişiklikler yapabiliriz. Başlık çubuğuna program adını yazabilir, program simgesini değiştirebilir, sağ tuş menüsünü ve ana menüyü değiştirebiliriz. Biz de buradaki MyBoard 1.6 başlık değişimini ve Play ana menü dalını silinmesini rahatlıkla yaptık. Programımızın son halinde değişiklik gerekli gibi zaten. Hakkında deyince Flash çıkması çok saçma.

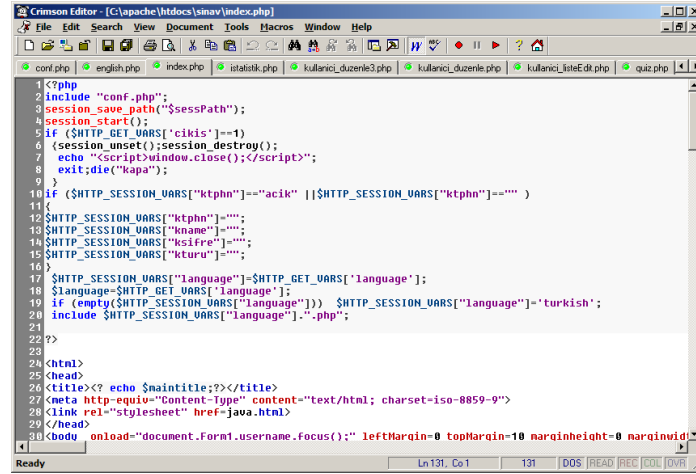
16. SiPro (Web tabanlı sınav programı geliştirme)

(Ornekler\Diger\sınav133c.zip)

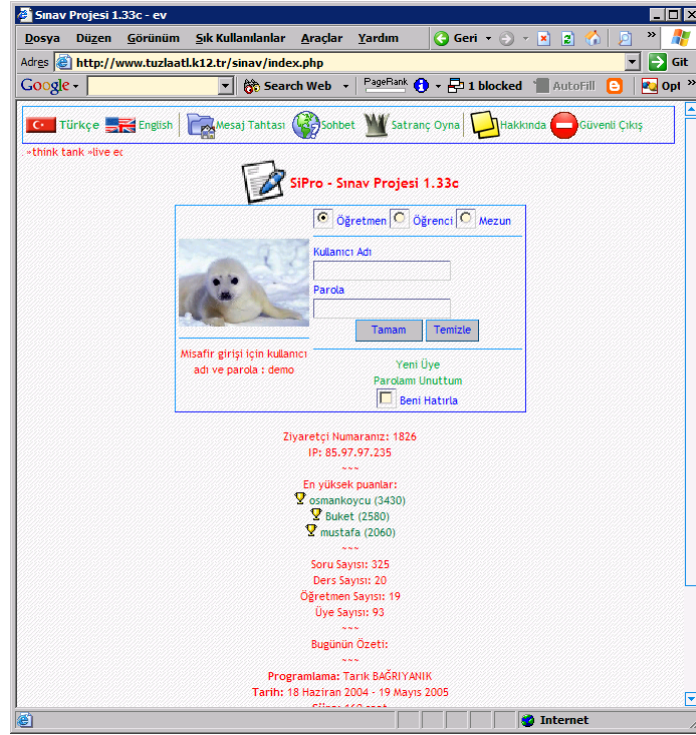
Yine PHP (Personal Homepage) ile yaptığım bu program ile de sınav otomasyonu yapıyoruz. KüPro kodlarını kullanarak yaptığım için iki projenin görünümüleri benziyor. Yaklaşık 6600 satır süren bu programda MySQL RDBMS (Remote Database Management System – Uzaktan veritabanı yönetim sistemi) veritabanı kullanmayı tercih ettim.



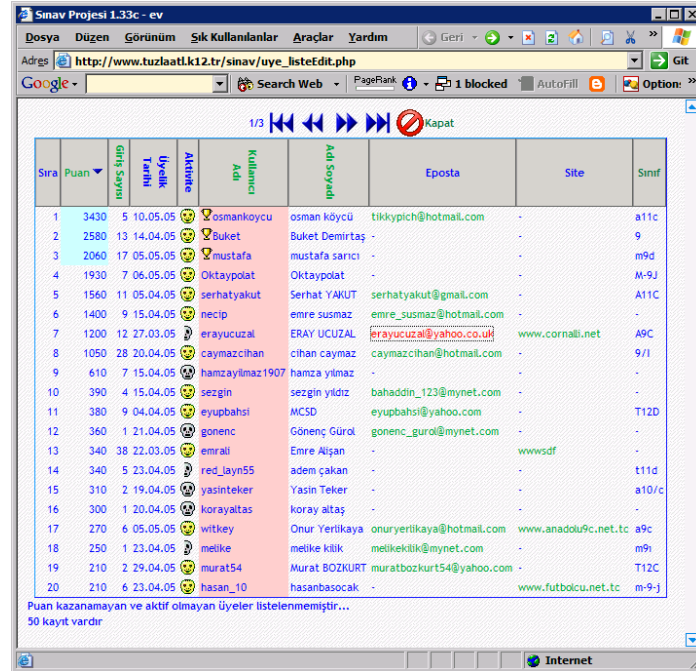
Resim 20a – phpMyAdmin ekranı



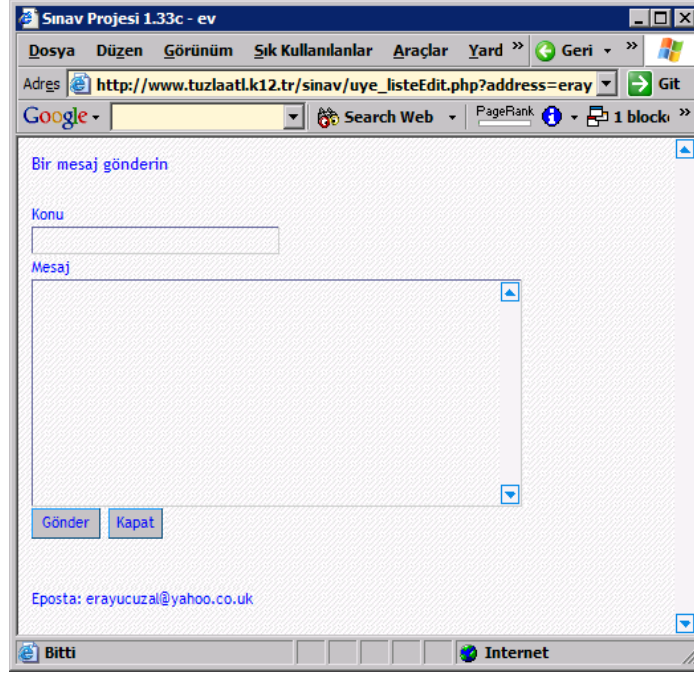
Resim 20b – Crimson metin düzenleyicisi



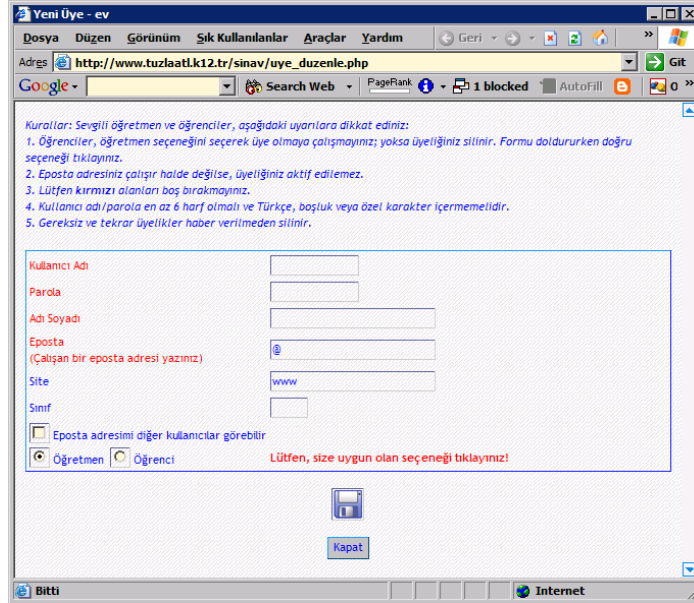
Resim 20c – SiPro açılış ekranı



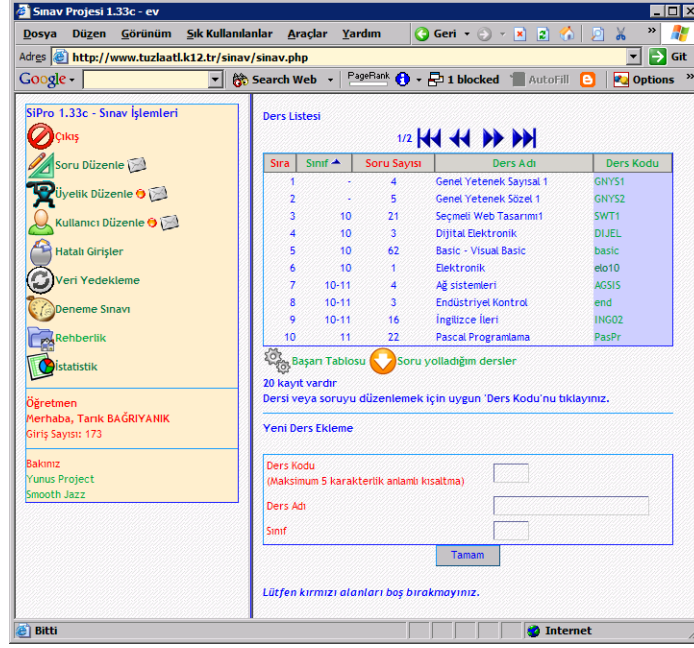
Resim 20d – Üye listesi



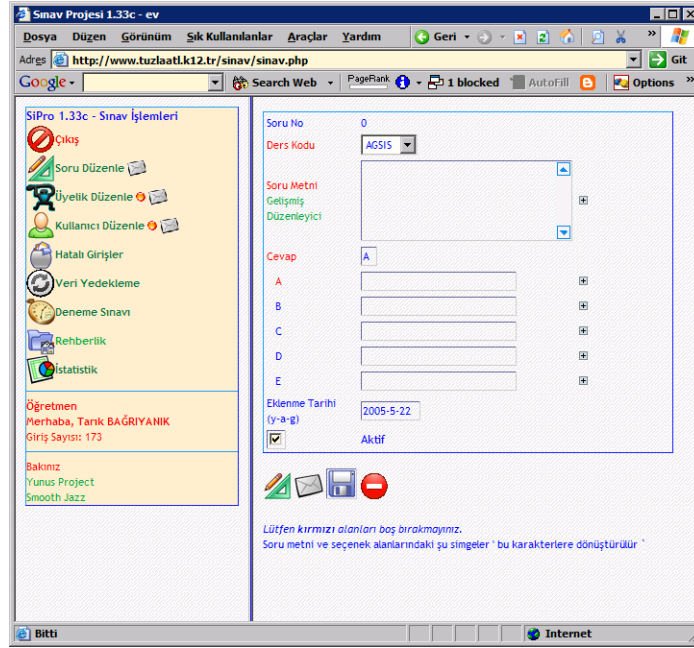
Resim 20e – Üye listesinde mail komutu ile üyeye elektronik mesaj atılabilir



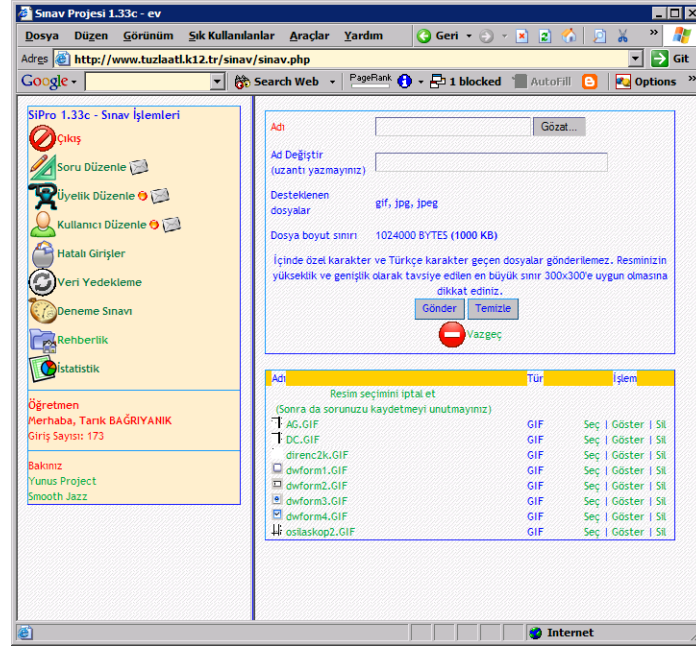
Resim 20f – Yeni üye ekleme



Resim 20g – Hatasız giriş yapmışız



Resim 20h – Soru düzenleme



Resim 20i – Sorulara resim ekleme

Okuldaki Linux türevi Gentoo ile nasıl çalışabilir yönünde tasarım yapmam gerekli idi. Bu sebeple evimdeki bilgisayara da benzer olan Mandrake 9.0/Suse 9.1 kurdum. Birkaç problem dışında kurulum kolaydı. Problemlerin en büyüğü kurulum sırasında bölümlendirme yaparken, tüm bilgilerimin silinmesi oldu. Aslında dikkatsiz bir şekilde kurulumdaki bir adımı geçince böyle oldu. Gelecek Linux 2.0 ve Mandrake 9.1 kurulum denemelerim tam istediğim gibi değildi. Özellikle MySQL kendiliğinden kurulmadığı için rpm dosyasını internetten ve kurulum CD'lerinin içinde aramak zorunda kaldım. Bu arada her Linux sürümü diğer Linux sürümlerinin *.rpm paketlerini okumayabiliyor. MySQL hizmetini Linux'da konsol ekranından "service -f mysql" komutu ile çalıştırıyoruz. Başlangıçta bu hizmetin çalışmasını ayarlamaya çalışsam da yapamadım. SuSe 9.1'de ise servisler görsel bir arabirimden çalıştırılıp durdurulabiliyorlar.

Birkaç gün uğraştıktan sonra Windows XP ve Mandrake 9.0 kurulu bilgisayarına kavuştum. Aslında ille de PHP'nin çalışması için Linux gerekli değil. Aynı ortamı Windows ile de oluşturabiliyoruz. Zaten kısa bir süre sonra kullanım kolaylıklarından dolayı Windows'a geri döndüm. Özellikle Kate metin editörü dışında aradığım yazı programını bulamadım. Kısa yolları hemen hemen Windows'un not defteri ile aynı olmasına rağmen alışamadım. Mozilla ve Konqueror tarayıcıları da maalesef Internet Explorer gibi kullanışlı değil. İleride Linux bu açıkları muhtemelen kapatacaktır. Şimdilik yaptığım değişiklikleri bazen Linux'da deniyorum.

Windows'da projenin çalışması için şu sırayı takip ediniz:

- 1) PHPTriad/Xampp paketini kurarak Apache ve MySQL'i çalıştırıyoruz.
- 2) Sınav133c.zip dosyasını "apache\htdocs" klasörüne "sinav" isminde klasör oluşturarak açınız.
- 3) MySQL'de veritabanı düzenlemek için en rahat ortam olan "<http://localhost/phpmyadmin>" adresini açalım. phpMyAdmin tablo oluşturmada ve düzenlemede en basit arabirimi sağlıyor. *duyuru* isminde yeni bir veritabanı açarak sınav klasöründeki "vt.sql" dosyasını SQL sekmesinde çalıştıralım. Başarı ile tabloların eklenmesi gerekli.
- 4) Windows dizinindeki php.ini dosyasındaki "register_globals = off" satırı olmasına dikkat edelim. Kütüphane projesinde ise "on" olmak zorundaydı. SiPro, birçok yönden KüPro'dan daha olgun.
- 5) Projeyi "<http://localhost/sinav>" diye açtığımızda, oturum kayıtlarını geçici bir dizinde dizininde sakladığımız için muhtemelen "c:\tmp" de sess_ef2b2b6ded... gibi dosyalar oluşacak. Daha sonra bunları rahatlıkla silebilirsiniz. Linux'da ise kök dizine yazma hakkı verilmez, o sebeple oturumlar "/tmp" gibi yerlere yönlendirilmelidir. Eğer kullanıcı ve üyeler "Güvenli Çıkış" ile çıkarılsa, kendi oturum dosyaları zaten siliniyor.
- 6) admin ile kullanıcı ve üye açabilir, soru ekleyebilir, sınav olabilir; kullanıcı yani öğretmenler soru ekleyebilir, sınav olabilir; üyeler yani öğrenciler sınav olabilir.
- 7) turkish.php ve english.php dosyalarında gerekli düzenlemeler yapılarak başka dillere destek veriliyor.

Kurulumu kendi ayarlarımıza göre değiştirmemiz gerekli. Bunun için projemizdeki "conf.php" dosyasına bakabiliriz. Örneğin MySQL'deki kullanıcı adı ve şifresini burada server olarak kullanmışız. Siz de kendi kullanıcı adı ve

şifresini belirlemeniz gerekebilir. Bu ayarları phpMyAdmin'den "mysql" veritabanından "users" tablosunda bulabilirsiniz.

php.ini'de de güvenlik ve performans amacı ile "register_globals = Off" yapıyoruz. Bu sebeple \$HTTP_SESSION_VARS, \$HTTP_GET_VARS, \$HTTP_POST_VARS, \$HTTP_COOKIE_VARS gibi komutlar kullanarak değişkenleri form veya adresten alabiliyoruz. Bazen aynı adı olan hem GET hem de POST edilmiş değişkenler oluyor. O zaman da empty() fonksiyonu ile nereden geldiğini belirlemeye çalışıyoruz.

Sizin veritabanının adı da duyuru olmak zorunda değil. Oturumun kaydedileceği klasörü de belirttikten sonra, son olarak projemizdeki "sinav/yollanan" isimli klasöre herkes için yazma hakkı verilmesi gerekiyor. Eğer ftp ile ana bilgisayara dosyaları atıyor iseniz, giriş yaptıktan sonra bu klasöre sağ tıklayıp, haklarını değiştirebilirsiniz. Linux'da da aynı şekilde hakları değiştirebilirsiniz, veya bir konsol ekranından "chmod" komutu verebilirsiniz.

Bunları düzenledikten sonra artık sınav projemiz sorunsuz çalışır hale geliyor. Öğrenci üyeler başvurdıkları anda aktif hale geldikleri halde, öğretmenlere daha sonra admin eposta atarak aktif olduklarını belirtiyor. Her yeni üye olduğunda admin'e eposta yollanması admin'i zor durumda bırakabilir. Hatta en çok yapılan hata da öğrencilerin yanlışlıkla öğretmen olarak başvurmaları. Şimdilik öğrenci öğretmen farkını çözmenin bir yolu yok. Yüz yüze, konuşarak onay alınması gerekebilir. Öğretmenin tek farkı soru oluşturma ve değiştirme hakkı olması. Öğrenciler de sınava girerek puanlarını artırmaya çalışıyorlar. Bu amaca göre hareket edersek, öğretmenlerin şifrelerini öğrencilere belli etmemeleri de iyi olur :)

Eposta adresi gizli olmayan öğretmenlere ve öğrencilere, adları geçen yerlerde (istatistik, başarı listesi, soru baskı önizlemesi), eposta atma penceresi kullanarak eposta yollanabiliyor.

Basitçe bir ay süreli ömrü olan bir çerez kullanarak, dil, son giriş yapanın adı, türü saklanabiliyor. "Beni Hatırla" seçeneğini iptal etmek için de, seçimi kaldırıp, giriş yapmak yeterlidir.

admin ise soru oluşturma, kullanıcı(öğretmen) ve üye(öğrenci) düzenleme silme yetkisine sahip. Öğretmen ve admin'in puanları derecelenmede etkili olmasa da, istatistikte girdikleri sınavlar belli olduğu için puanlarını görebilirler.

Sorular ortak bir havuzda tutulduğu için, örneğin Edebiyat öğretmeni diğer derslerdeki Türkçe yazım hatalarını düzeltebilir. Böylece gittikçe daha güzel sorular ortaya çıkabilir. Hem de kişiye özel ders olmuyor. Pascal bilen bilgisayar öğretmenleri benzer soruları bir yerde toplayıp, ortak sınav gibi hazırlık yapabilirler. Öğrenciler öğretmene göre değil, konularına göre ders çalışırlar.

Veri girişi uzun sürüyor, bu sebeple, soruları bilgisayar başında o anda üretmek yerine, önceden hazırlayıp, girmek daha çabuklaştırıyor. Ama Excel'den şu verileri al gibi bir özellik olsaydı, iyi olurdu. Benzer şekilde soruları kaydedip daha sonra çalışma imkanı "Soru Önizleme" ile yapılsa da, soruları gerçek sınavlarda kullanmak isteyen öğretmenler biraz uğraşmaları gerekiyor. Yani belli soruları numaralandırıp Word'e aktarsaydık çok faydalı olurdu. Soru önizlemesinde soruları Ctrl+A ve Ctrl+C ile kopyalayıp, Word'e "Özel Yapıştır...*Biçimlendirilmiş Metin(RTF)" olarak yapıştır denilince biraz da olsa işe yarar şeyler elde edebiliyoruz. Yapıştırma işleminde HTML biçimi seçilirse resimler de yapıştırılabilir.

Sadece belli bir haftada sınav sorularının aktif olması istenirse, "Dersi Kilit" seçeneği kullanılıyor. Böylece o dersin tüm soruları pasif yapılarak, öğrencilerin o sınava ulaşmaları engelleniyor. Ne zaman ne olacağı ana sayfadan duyurulabilirdi. Şimdilik bu konu ile ilgili haberleşme veya öğrencilere not bırakma yok.

Öğrenciler anlamadıkları, bilemedikleri soruları Google'dan arayabilir. Çok uzun sorularda pek faydalı olmasa da öğrenci arama metnini azaltıp tekrar yazarak arama yapabilir. Cevapları sınav bitiminde veya soru cevaplanırken hiçbir yerde görülemiyor. Soru dökümüne de cevapları eklemedim. Bu sayede öğrenci araştırma yapmaya da yönlendiriliyor. Bilemediği soruları internetten arayabilir veya öğretmenine sorabilir.

Soru ve derslerin zorluk derecesi de zamanla cevaplandıkça, sınavlar yapıldıkça, değişiyor. Genellikle tek seçenekli sorular en zorları. Bir harf hatası olursa yanlış oluyor. Soru hazırlarken eğer tek seçenek de kullanılsa, cevap kısmına bir harf girilmesi gerekir. Eğer örneğin B seçeneği boş bırakılıp C, D yazılsa bile, sınav sırasında B seçeneği gösterilmeyip, C, D olarak devam ediyor. Yani soru hazırlanırken boş bırakılan seçeneklerin harfleri düzeltilmiyor. A, C, D olarak gidiyor.

En iyisi çoktan seçmeli sorular. En fazla 5 seçeneğe kadar seçenek sunarak soru hazırlanabilir. Soru metni ve seçeneklere resim de eklenebiliyor. Bunun için "sinav/yollanan" klasörü kullanılıyor. Aynı resim başka sorularda da kullanılabilir. İstenirse hem yazı hem de resim birlikte olabilir. Desteklenen dosya türleri JP*G ve GIF. 1 MB sınırı çok fazla olsa bile tahminim kimse 1260*1024 gibi resim yollamaz. Zaten resim ekranda en iyi 300*300 görüntüleniyor.

İleriki aşamalar olarak, sınavların süreli olması, daha estetik görünüm, isteyenlere kolaylık derecesine göre sınav seçeneği sunma gibi şeyler yapılabilir. Flash ile de sorulara animasyon desteği verilebilir.

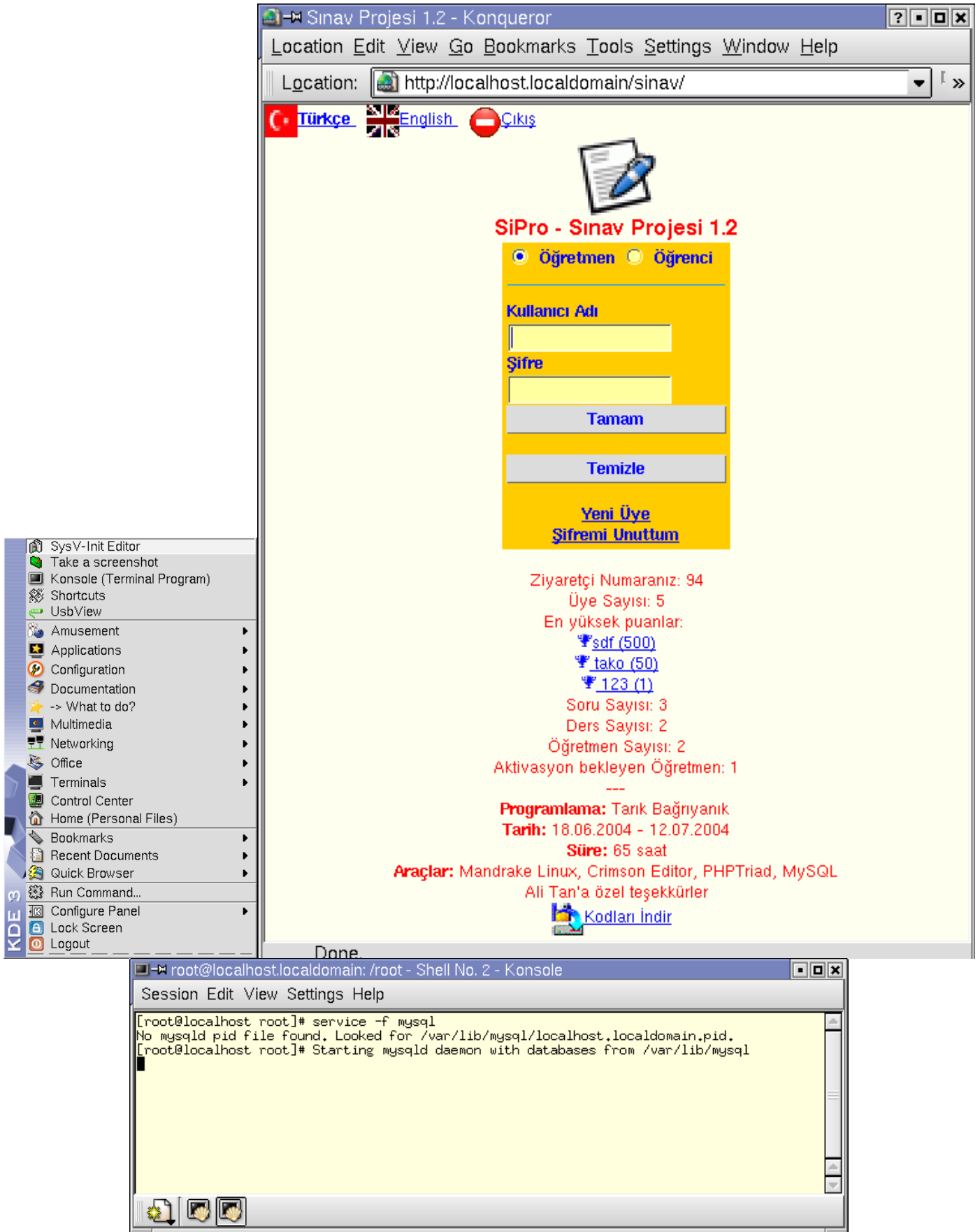
Bu proje ile ilk kez neredeyse tüm platformlarda çalışabilecek bir program yapmamız oldu. JavaScript projeleri de platform'dan bağımsız olsa da gelişmiş programlar PHP ile kolayca halledilebiliyor. DotNET ile de Microsoft bu global programcılığı sağlamaya çalışıyor. Ama şimdilik PHP bedava olması, hızlı ve kolay öğrenilmesi bakımından en çok tutulandı. Hem ASP hem de PHP destekli az da olsa ana bilgisayar(hosting firması) bulabilirsiniz. Siz de kendi bilgisayarınızı host – ana bilgisayara dönüştürebilirsiniz. Normalde port numaranız 80 olduğu için, örneğin; apache için 81, IIS için 80 numaralı portu seçerek, aynı anda ASP ve PHP çalıştırabilirsiniz. <http://localhost> ASP için, <http://localhost:81> PHP için olur. Eğer sabit IP'li bir bağlantınız var ise ve her zaman çalıştırılabileceğiniz bir bilgisayarınız var ise kendi web server'inizi yapabilirsiniz. Linux + Apache + PHP + MySQL güçlü ve bedava bir konfigürasyon. Windows + IIS + MSSQL ise biraz pahalı ve güvenlik açıkları bol bir sistem. Aslında Linux içinde güvenlik sorunlarından bahsedebiliriz fakat, internetten rahatlıkla küçük güncellemeler ile bunların üstesinden gelebilirsiniz.

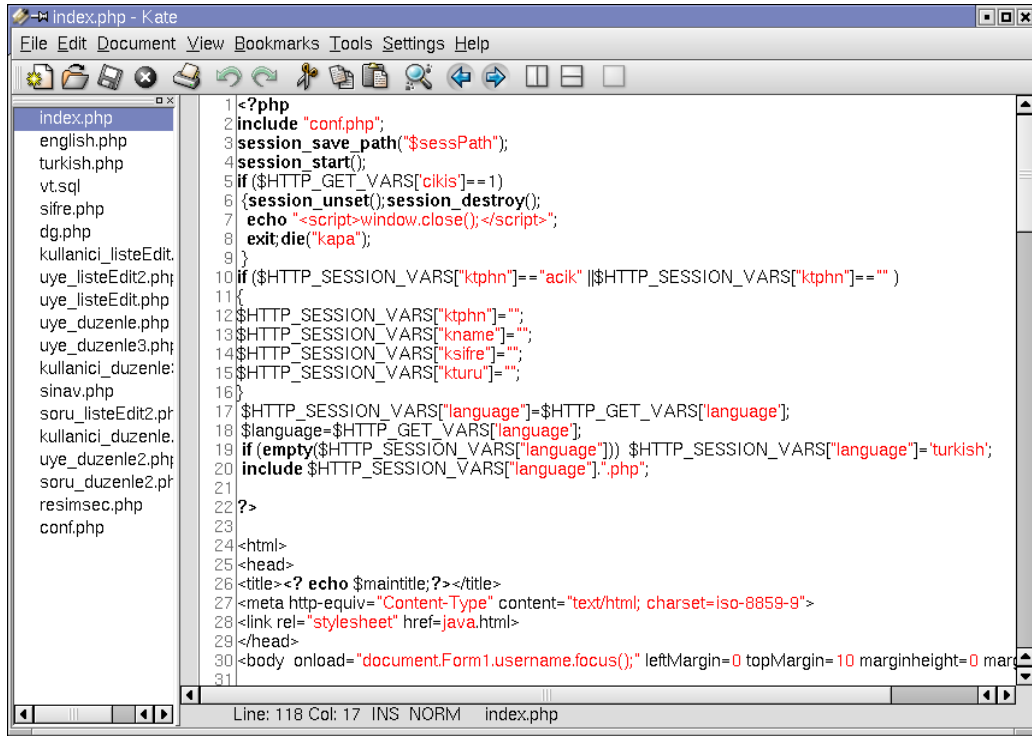
Sonuç olarak tek noktadan çalışarak, tüm dünyada geçerli olan, aynı anda çoklu kullanıcı girişi yapılabilen, bir eğitim ağı kurulmuş oluyor. Makine ve işletim sistemine bağımlı olmayan, hızlı, estetik, genişletilebilir bir proje. Gelişmiş bir cep telefonundan bile çalışabilir, en eski bilgisayarda da çalıştığını görmek sürpriz olmaz.

Apache, PHP ve MySQL üzerinde çalışan moodle.com (45 MB kurulmuş hali) ile atutor.ca (4.5 MB kurulmuş hali) projelerini bedava açık kaynak kod LMS (Learning Management System – Öğrenme Yönetim Sistemi) deneyebilirsiniz. Özellikle moodle çok gelişmiş. Java alternetifimiz de olat.org projesidir. OLAT kurulumunda kendi server programı ile geliyor. Yani tam bir paket (50 MB kurulmuş hali). İçinde Tomcat5 ve HSQL veritabanı var. Zürih Üniversitesi'nde yapılmış. Administrator yani yönetici tüm siteye bakım yapma, üyelikleri kontrol etme gibi işleri varken, öğretmenler kurs açıyor, öğrenciler de diledikleri kurslara kayıt oluyor. İster sohbet, isterse forumlara da katılabilir. Aralarında mesajlaşma yolu ile de bağ kuruyorlar.

SiPro'da da (3 MB kurulmuş hali) buna benzer işlevler varsa da, konu anlatımı ve görsel bütünlük konusunda çok eksiklerimiz var. Resim düzenleyici de ekleme isterdim. Öğretmen soruları eklerken, soru metnine resim yaparak da daha güzel anlatabilir. Metin düzenlemede bedava Protopad (<http://www.xtort.net/xtort/protopad.php>) ve fckeditor.net yardımcı olabilirdi. WYSIWYG metin düzenleme kolay ama, resim işlemede alternatif bulamadım.

E-eğitimde içerik hazırlamak için en iyi yol Flash. Çoğu bilgisayarda Flash Player yüklü veya internetten kurması çok kolay. Powerpoint ile de ders anlatımı yapılabilse de yaygın değil. Uzun belgeler genelde *.pdf olarak site içine ekleniyor. Her ne kadar Java ve JavaScript ile dinamik görüntü yapabilsen de Flash kadar etkili değil.





Resim 20j – Linux’den birkaç görüntü

17. Denklemler (Excel ile denklem çözümü)

(Ornekler\Diger\ExceldeDenklemler.zip)

Byte 1998 Mart sayısından yaptığım bu örnekte, 2 yöntemle "n. dereceden bir bilinmeyenli denklem", 2 yöntemle de "birinci dereceden n bilinmeyenli denklem" çözülebiliyor. Microsoft Excel'in bir eklentisi olan "Çözücü Eklentisi" kurulu olmalıdır. İsterseniz istatistik çalışmaları için de "ToolPak Çözümleyicisini" yükleyin.

E6			=	=D6^3+6*D6^2+3*D6-10		
	A	B	C	D	E	F
1						
2						
3						
4						
5				x	y	
6			ornek1=	1	-1,8E-07	
7			ornek2=	-10	-440	
8						

E6			=	=D6^4+D6^3-19*D6^2+11*D6+30			
	A	B	C	D	E	F	G
1							
2							
3							
4							
5				x	y		
6			ornek1=	-1	9,32E-07		
7			ornek2=	1	24		
8			ornek3=	8	3510		
9							

D8			=	=x+y+z-4			
	A	B	C	D	E	F	G
1			x+y+z=4				
2			2x+y-z=-2				
3			3x+4y-z=1				
4							
5				x	y	z	
6			ornek=	0	1	3	
7							
8			Çözüm=	1E-06			
9				0			
10				-2,2E-15			

Yardım dosyasında çözümleme aracı şu şekilde tanımlanmış:

"Microsoft Excel, karmaşık istatistiksel ve mühendislik çözümlemeleri geliştirdiğinizde adımları kaydetmek için kullanabileceğiniz (ToolPak Çözümleyicisi adında) bir veri çözümlemesi kümesi sağlar. Her çözümlemenin verilerini ve parametrelerini siz sağlarsınız; araç, uygun istatistiksel veya mühendislik makro işlevlerini kullanır ve sonra sonuçları bir çıktı tablosunda görüntüler. Bazı araçlar, çıkış tablolarına ek olarak, grafikler de oluşturur."

Yani çözümleme Anova, Korelasyon, Kovaryans, Tanımlayıcı İstatistik, Üstel Düzeltme, Fourier Çözümlemesi, Hareketli ortalama, Örnekleme, Regresyon, t-test, z-test, Histogram gibi araçlarla da yapılabilir. Buradaki "n. dereceden bir bilinmeyenli denklem" ve "birinci dereceden n bilinmeyenli denklem" için bu araçlara gerek kalmadan basitçe "Çözücü" ile işimizi hallediyoruz.

D9			=	=x+2*y-2*z+3*t-5				
	A	B	C	D	E	F	G	
1			x+2y-2z+3t=5					
2			x+3y+2z+5t=11	Birinci dereceden n bilinmeyenli denklem				
3			x+2y-z+t=6					
4			x+4z+3t=6					
5								
6				x	y	z	t	
7			ornek=	3	2	1	0	
8								
9			Çözüm=	0				
10				0				
11				0				
12				0				

Resim 21a – Excel sayfalarının görünüşleri

Çözücü Parametreleri

Hedef Hücre:

Eşittir: ☐ En Büyük ☐ En Küçük ☒ Değer:

Değişen Hücreler:

Kısıtlamalar:

-
-

Çözücü Parametreleri

Hedef Hücre:

Eşittir: ☐ En Büyük ☐ En Küçük ☒ Değer:

Değişen Hücreler:

Kısıtlamalar:

-
-

Çözücü Parametreleri

Hedef Hücre:

Eşittir: ☐ En Büyük ☐ En Küçük ☒ Değer:

Değişen Hücreler:

Kısıtlamalar:

Çözücü Parametreleri

Hedef Hücre:

Eşittir: ☐ En Büyük ☐ En Küçük ☒ Değer:

Değişen Hücreler:

Kısıtlamalar:

Resim 21b – Çözücü ayarları

Sonuçları Yanıt Raporu, Duyarlılık Raporu ve Sınırlamalar Raporu olarak görebiliriz. Çözmeden önce Seçenekler menüsünden de Yineleme, Tolerans, Yakınsama ve Duyarlılık gibi ayarları yapabiliriz. Formülü girdikten sonra yeşil olarak renklendirdiğim hücelere rakam girerek sonucu deneme imkanımız var. Ama bu denemeleri neyseki Excel otomatik olarak bitiriyor. Yani sonuçlar 0 olana kadar belli aralıkta denemeler yapıyor.

Formüllerde gördüğünüz x, y ve z gibi harfleri hücelere isim vererek yapıyoruz. Mesela D8 hücresi x değerine denk gelecek ise, "Ekle*Ad*Tanımla..." menüsünde isim verebiliriz. Bu sayfada istediğiniz yerde formüllerde D8 yerine x yazabilirsiniz. İstedığımız zaman da formül çubuğunun solundaki "Ad Kutusu"na x yazarak o hücreye gidebiliriz.

Çözücü parametreleri pencerelerinde hücre isimleri \$ simgeleri ile yazılmalıdır. Böylece referansı sabit olan hücre isimleri elde ediyoruz.

18. ASP.NET uygulaması (Online Telefon Defteri)

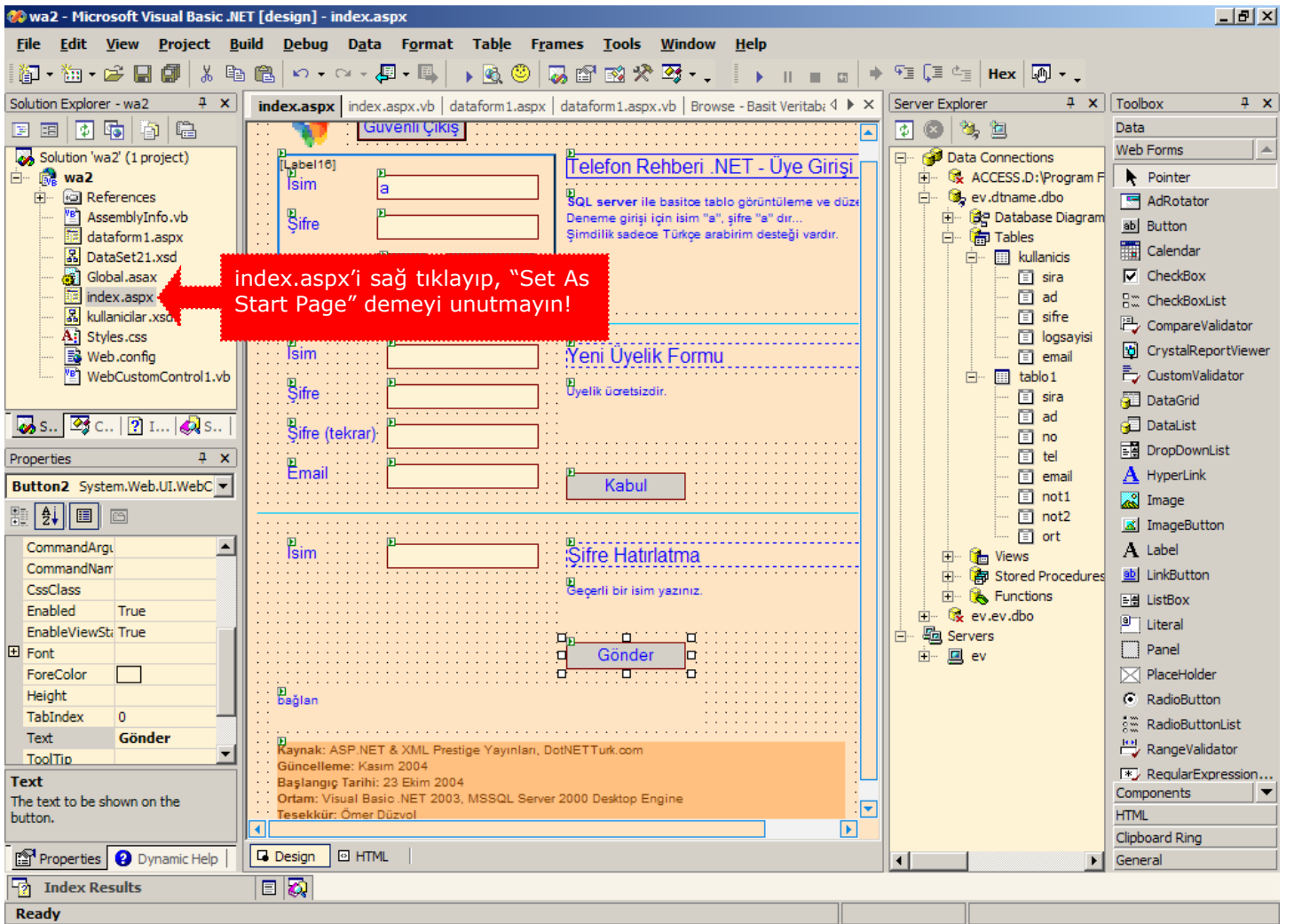
(Ornekler\Diger\wa211.zip)

Microsoft, .NET ile programcılar için en rahat ortamı sunmaya çalışıyor. Gerçekten de artık kökten bir değişim gibi gözüküyor ama, tüm programcılara hitap eden bir arabirimi var. Resimde görüldüğü gibi Visual Basic'in IDE'sine çok benziyor. C#, C++, Basic, J# veya Delphi fark etmeden tek bir IDE (Integrated Development Environment – Entegre Geliştirme Ortamı) ile ister Windows, ister Web uygulaması, isterse de el bilgisayarı (PalmPC) programları hızlı bir şekilde yapılabilir. Unix desteği de var. XML ağırlıklı olmasına rağmen de Access hala en çok kullanılan veritabanı. Daha etkili ve kapasiteli veritabanı uygulamaları da MSSQL Server ile yapılabilir.

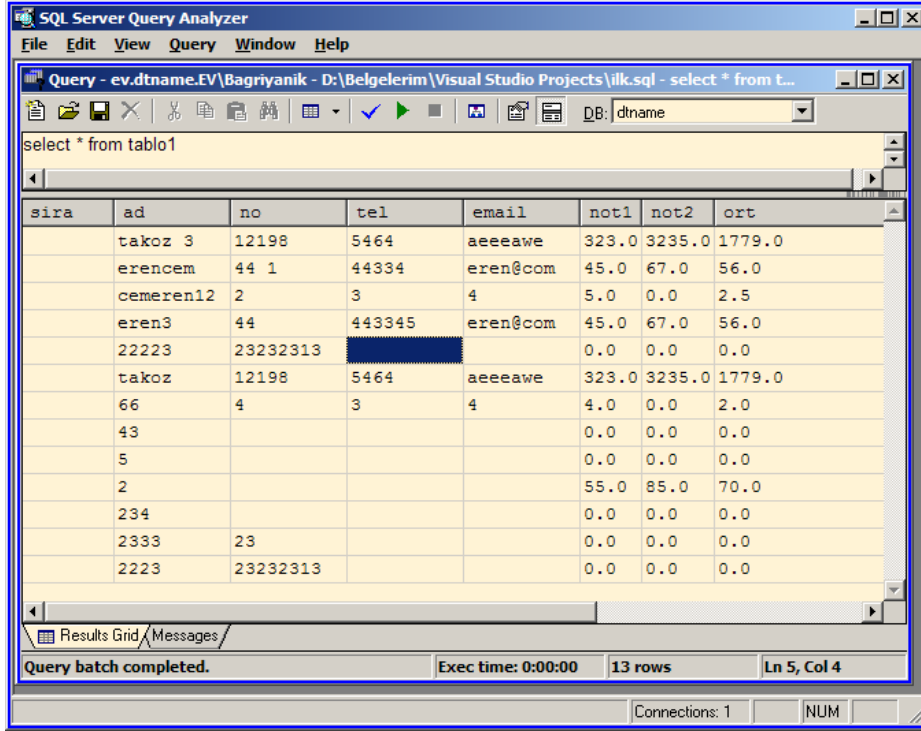
Yaklaşık 40 saat (700 satır) uğraştıktan sonra Telefon Rehberi programımız olgunluğa erişti. Altyapı olarak Windows XP, ASP.NET 2003, Internet Information Services (IIS), MSSQL Server Service Desktop Edition 8.0 kullandım.

IIS'yi Windows CD'si kullanarak Program Ekle Kaldır'dan eklemeyi unutmayın. "C:\Inetpub\wwwroot" klasörü bizim ana dizinimiz olacak. Visual Basic.NET'de yeni proje diyerek, "ASP.NET Web Application" seçince ana dizinin altında wa2 gibi uygulamamız oluşur. Son aşamada, sitemizi "<http://localhost/wa2/index.aspx>" diye internet gezginimizden görebiliriz.

ASP.NET kurulduktan sonra otomatik oluşan ASPNET kullanıcısına, yetki vermemiz gereklidir. "Başlat*Ayarlar*Denetim Masası*Yönetimsel Araçlar*Bilgisayar Yönetimi*Sistem Araçları*Yerel Kullanıcılar ve Gruplar*Users" dan ASPNET seçilerek Üyelik kısmına Administrators, VS Developers ve Users ekliyoruz. Aslında Administrator hakkı vermek çok tehlikelidir, ama şimdilik öğrenmek için kendi bilgisayarımızda bu ayarı yapabiliriz.



Resim 22a – VB ASP.NET tasarım penceresi



Resim 22b – SQL Server Query Analyzer ile SQL denemelerimiz

Kullanılacak tabloları SQL Server'a "dtname" veritabanı altına *tablo1* (telefon bilgileri), *kullanicis* (kullanıcı bilgileri) oluşturuyoruz. Tasarım penceresindeki "Server Explorer" menüsüne sağ tıklayıp "Add Connection..." ile dtname eklenir.

Kullanicis tablosu yapısı:

sira	char	10
ad	char	10
sifre	char	10
logsayisi	int	4
email	char	10

Tablo1 yapısı:

sira	char	10
ad	char	10
no	char	10
tel	char	10
email	char	10
not1	float	8
not2	float	8
ort	float	8

Formula bilgisi (([not1] + [not2]) / 2)

dtname eklendikten sonra da Server Explorer menüsündeki dtname'ye "Tables*New Table" ile tabloları ekleyiniz. Aslında dtname içine otomatik olarak tabloları oluşturan SQL komutu da (CREATE TABLE) verebilirdik. SiPro'da MySQL için "vt.sql" yapmıştık hatırlarsanız. Tam sürüm MSSQL Server'da bu imkan vardır.

index.aspx ve dataform1.aspx olmak üzere iki sayfadan oluşan sitemiz var. PHP'deki gibi JavaScript ile uyarı (alert) ve onay (prompt) kısımlarını hallediyoruz. Sytles.css içinde de tüm sitenin stil ayarları var.

Veri Bağlantısı Özellikleri

Sağlayıcı Bağlantı Gelişmiş Tümü

SQL Server verisine bağlanmak için aşağıdakileri belirleyin:

1. Sunucu adını seçin veya girin:
2. Sunucuda oturum açmak için bilgileri girin:
☒ Windows NT tümleşik güvenliğini kullan
☐ Belirli bir kullanıcı adı ve parolası kullan:
 Kullanıcı adı:
 Parola:
☐ Boş parola ☐ Parola kaydetmeye izin ver
3. ☒ Sunucudaki veritabanını seçin:

☐ Veritabanı adı olarak bir veritabanı dosyası iliştiir:

 Kullanılacak dosya adı:

Resim 22c – dname veritabanı

Tablo Sayfası - Microsoft Internet Explorer

Adres: <http://localhost/wa2/dataform1.aspx?filtrele=23>

Ana Sayfa

Kullanıcı: a

Sıralama: ad
Yön: ASC
İşlemler: sıralama(çift yön), listeleme(10'arlı sayfalama), silme(onaylı), ekleme, güncelleme, arama, filtreleme (ayn bir tabloda), hesaplanan alanlar(ayn tabloda), kullanıcı girişi (kullanıcı ekleme, şifre hatırlatma)
(*) tam bilmedi
Kayıt sayısı:13

Seçimi İptal Et/Tazele Arama ☒ Sıralama Yönü: Artan Filtre metni: "23"

Kaydet Yeni Sil

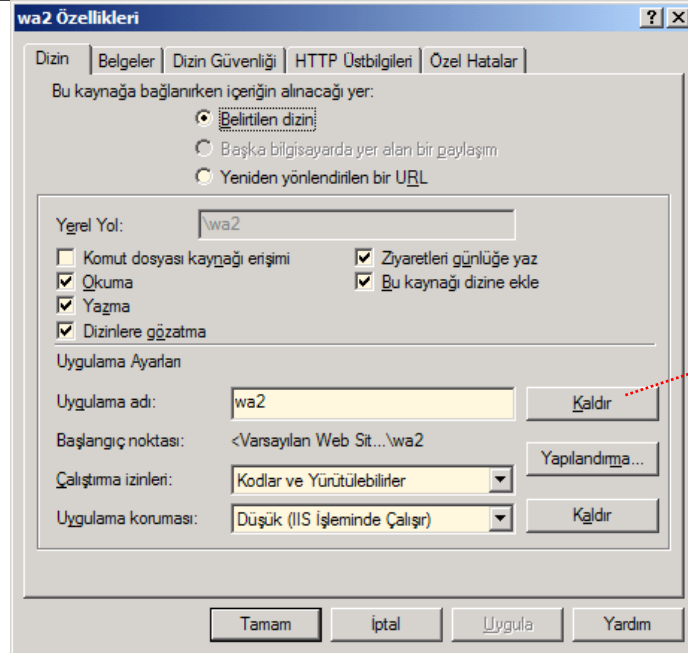
Önceki Sonraki

	Ad	No	Telefon	Email	Not1	Not2
Seç	2				55	85
Seç	22223	23232313			0	0
Seç	2223	23232313			0	0
Seç	2333	23			0	0
Seç	234				0	0
Seç	43				0	0
Seç	5				0	0
Seç	66	43	4		0	0
Seç	cemeren12	23	4		5	0
Seç	eren3	44443345	eren@com		45	67
Seç	İsimler					

Arama Tablosu

ad	no	tel	email	not1	not2	ort
22223	23232313			0	0	0
2223	23232313			0	0	0
2333	23			0	0	0
234				0	0	0

Local intranet



Resim 22d – Projenin çalışmış hali ve birkaç minik ayar daha

Bu projede çalıştırmayı başardığımız veritabanı kayıt işlemleri; sıralama (çift yön), listeleme (10'arlı sayfalama), silme (onaylı), ekleme, güncelleme, arama, filtreleme (ayrı bir tabloda), hesaplanan alanlar (ayrı tabloda), kullanıcı girişi (kullanıcı ekleme, email ile şifre hatırlatma).

?1? Eğer şu şekilde bir hata oluşmuş ise:

Cannot open database requested in login 'dtname'. Login fails. Login failed for user 'EV\ASPNET'.

Çözüm: Komut isteminde (Başlat*Çalıştır*cmd) şu adımları yapınız:

```
C:\>osql -E
1> use master
2> exec sp_grantlogin 'EV\ASPNET'
3> go
Granted login access to 'EV\ASPNET'.
1> use dtname
2> exec sp_grantdbaccess 'EV\ASPNET'
3> go
Granted database access to 'EV\ASPNET'.
1> exec sp_addrolemember 'db_datareader','EV\ASPNET'
2> go
'EV\ASPNET' added to role 'db_datareader'.
1> quit
C:\>_
```

EV yerine kendi bilgisayar adınızı yazmayı unutmayın. dtname yerine de kendi yaptığınız veritabanınız olmalı.

??? Eğer şu şekilde bir hata oluşmuş ise:(UPDATE yerine SELECT, INSERT veya DELETE de olabilir)

UPDATE permission denied on object 'Tablo1', database 'dtname', owner 'dbo'.

Çözüm: Üstteki hatadaki gibi adımları izleyerek, 'db_datareader' yerine 'db_datawriter' yapınız.

Belki C: sürücüsüne yazma hakkı yoktur. Yani C\$ şeklinde. Sürücüye sağ tıklayıp paylaşım açmanız gerekebilir. Hata hala oluşuyor ise [google.com](https://www.google.com)'a hata mesajınızı yazıp çözüm arayabilirsiniz. VB.NET 2005 veya sonraki versiyonlarda veritabanı programcılığı muhtemelen çok daha kolay olacaktır. Aşağıdaki kodlara bakınca, fazla gibi gezebilir. Kodlarla programa hakimiyet devam ediyor.

index.aspx'in kodları:

```
Imports System.Web.Security
Imports System.Web.Mail
Public Class WebForm1
    Inherits System.Web.UI.Page
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
    Dim cs, cv As DataView
    Dim ds As New DataSet
    Dim sifre As String = ""
    Protected WithEvents Image1 As System.Web.UI.WebControls.Image
    Protected WithEvents TextBox1 As System.Web.UI.WebControls.TextBox
    Protected WithEvents TextBox2 As System.Web.UI.WebControls.TextBox
    Protected WithEvents Submit1 As System.Web.UI.HtmlControls.HtmlInputButton
    Protected WithEvents Label2 As System.Web.UI.WebControls.Label
    Protected WithEvents Label3 As System.Web.UI.WebControls.Label
    Protected WithEvents DataGrid1 As System.Web.UI.WebControls.DataGrid
    Protected WithEvents Label4 As System.Web.UI.WebControls.Label
    Protected WithEvents TextBox3 As System.Web.UI.WebControls.TextBox
    Protected WithEvents TextBox4 As System.Web.UI.WebControls.TextBox
    Protected WithEvents TextBox5 As System.Web.UI.WebControls.TextBox
    Protected WithEvents TextBox6 As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label5 As System.Web.UI.WebControls.Label
    Protected WithEvents Label6 As System.Web.UI.WebControls.Label
    Protected WithEvents Label7 As System.Web.UI.WebControls.Label
    Protected WithEvents Label8 As System.Web.UI.WebControls.Label
    Protected WithEvents Label9 As System.Web.UI.WebControls.Label
    Protected WithEvents Label10 As System.Web.UI.WebControls.Label
    Protected WithEvents Label11 As System.Web.UI.WebControls.Label
    Protected WithEvents TextBox10 As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label13 As System.Web.UI.WebControls.Label
    Protected WithEvents Label12 As System.Web.UI.WebControls.Label
```

```

Protected WithEvents Button2 As System.Web.UI.WebControls.Button
Protected WithEvents Button1 As System.Web.UI.WebControls.Button
Protected WithEvents Label14 As System.Web.UI.WebControls.Label
Protected WithEvents Label15 As System.Web.UI.WebControls.Label
Protected WithEvents Label16 As System.Web.UI.WebControls.Label
Protected WithEvents Button3 As System.Web.UI.WebControls.Button
Dim dr As DataRow

Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init
    'CODEGEN: This method call is required by the Web Form Designer
    'Do not modify it using the code editor.
    FormsAuthentication.SetAuthCookie(TextBox1.Text, False)
End Sub
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    'Put user code to initialize the page here
    Dim s As String = "data source=localhost;initial catalog=dtname;integrated security=true"
    Dim conn As New SqlClient.SqlConnection(s)
    Try
        conn.Open()
        If conn.State = ConnectionState.Open Then
            Label4.Text = "<em>Veritabanına bağlantı var...</em> "
        Else
            Label4.Text = "<em><font color=red>veritabanına bağlantı yapılamıyor!...</em> </font>"
        End If
        conn.Close()
    Catch ex As Exception
        Label4.Text = "<em><font color=red>Veritabanına bağlantı yapılamıyor!...</em></font>"
    End Try
    Application.Lock()
    Application("Visitors") += 1
    Application.Unlock()
    Label4.Text &= "<br>Ziyaretci Sayısı= " & Application("Visitors")
End Sub
Function bul2(ByVal ad As String)
    Dim email As String = "" 'bulunamazsa boşluk olur
    Dim s As String = "data source=localhost;initial catalog=dtname;integrated security=true"
    Dim conn As New SqlClient.SqlConnection(s)
    Dim dp1 As New SqlClient.SqlParameter("@ad", SqlDbType.Char)
    Dim da As New SqlClient.SqlDataAdapter("select * from kullanicis where ad=@ad", conn)
    Dim ds As New DataSet
    dp1.Value = ad
    da.SelectCommand.Parameters.Add(dp1)
    conn.Open()
    sifre = ""
    da.Fill(ds, "kullanicis")
    If ds.Tables(0).Rows.Count > 0 Then
        email = ds.Tables(0).Rows(0)(4)
        sifre = ds.Tables(0).Rows(0)(2)
    End If
    conn.Close()
    Return email
End Function
Function bul(ByVal ad As String, ByVal sifre As String)
    Dim tamam As Boolean = False
    Dim s As String = "data source=localhost;initial catalog=dtname;integrated security=true"
    Dim conn As New SqlClient.SqlConnection(s)
    Dim dp1 As New SqlClient.SqlParameter("@ad", SqlDbType.Char)
    Dim dp2 As New SqlClient.SqlParameter("@sifre", SqlDbType.Char)
    Dim da As New SqlClient.SqlDataAdapter("select * from kullanicis where ad=@ad and sifre=@sifre", conn)
    Dim ds As New DataSet
    dp1.Value = ad
    dp2.Value = sifre
    da.SelectCommand.Parameters.Add(dp1)
    da.SelectCommand.Parameters.Add(dp2)
    conn.Open()
    da.Fill(ds, "kullanicis")
    If ds.Tables(0).Rows.Count > 0 Then
        tamam = True
    End If
    conn.Close()
    Return tamam
End Function
Private Sub Submit1_ServerClick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Submit1.ServerClick
    If ((TextBox1.Text <> "") And (TextBox2.Text <> "")) Then

```

```

        If bul(TextBox1.Text, TextBox2.Text) Then
            Application.Lock()
            Application("Ad") = TextBox1.Text
            Application("Sifre") = TextBox2.Text
            Application.Unlock()

            Response.Redirect("dataform1.aspx")
            'TextBox1.Text = "olur"
        End If
    Else
        TextBox1.Text = "hata"
    End If
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim email As String
    email = bul2(TextBox6.Text)
    If TextBox6.Text <> "" And TextBox5.Text <> "" And _
        TextBox6.Text <> "admin" And _
        TextBox4.Text <> "" And TextBox3.Text <> "" And TextBox4.Text = TextBox5.Text And _
        email = "" Then
        Dim s As String = "data source=localhost;initial catalog=dbname;integrated security=true"
        Dim conn As New SqlConnection(s)
        Dim dp1 As New SqlParameter("@ad", SqlDbType.Char)
        Dim dp2 As New SqlParameter("@sifre", SqlDbType.Char)
        Dim dp3 As New SqlParameter("@email", SqlDbType.Char)
        Dim da As New SqlDataAdapter
        Dim ds As New DataSet
        da.InsertCommand = New SqlCommand
        da.InsertCommand.Connection = conn
        da.InsertCommand.CommandText = "INSERT INTO kullanicis (ad,sifre, email) VALUES (@ad, @sifre, @email)"
        dp1.Value = TextBox6.Text
        dp2.Value = TextBox5.Text
        dp3.Value = TextBox3.Text
        Label12.Text = "-" & da.InsertCommand.CommandText
        da.InsertCommand.Parameters.Add(dp1)
        da.InsertCommand.Parameters.Add(dp2)
        da.InsertCommand.Parameters.Add(dp3)
        conn.Open()
        da.InsertCommand.ExecuteNonQuery()
        conn.Close()
        Label12.Text = "Üyelik ücretsizdir.<br>Üyelik kabul edilmiştir."
        TextBox6.Text = ""
        TextBox5.Text = ""
        TextBox4.Text = ""
        TextBox3.Text = ""
    Else
        Label12.Text = "Üyelik ücretsizdir.<br><font color=red>Üyelik formunda bir <b>hata</b>
oluşturdu!<br>Aynı isimde bir kullanıcı olabilir veya formda eksiklikler var...</font>"
    End If
End Sub

Function SendMailEx(ByVal toAdress, ByVal toName, ByVal subject, ByVal Body)
    Dim email As New System.Web.Mail.MailMessage
    email.To = toAdress
    email.From = "SenderAddress"
    email.Body = Body
    email.Subject = subject
    email.BodyFormat = Web.Mail.MailFormat.Text
    System.Web.Mail.SmtpMail.SmtpServer = "localhost"
    'System.Web.Mail.SmtpMail.Send(email) 'şimdilik çalışmıyor
End Function

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Dim email As String
    email = bul2(TextBox10.Text)
    If email <> "" Then
        SendMailEx(email, "Telefon Rehberi", "Şifre Hatırlat", "Şifreniz: " & sifre)
        Label14.Text = "<b><font color=blue>Şifreniz email adresinize gönderildi.</font></b>"
        'Label14.Text = email
    Else
        Label14.Text = "<b><font color=red>Üye bulunamadı!</font></b>"
    End If
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

```



```

Application("Ad") = ""
Application("Sifre") = ""
Response.Write("<script>self.close();</script>")
End Sub

Private Sub InitializeComponent()

End Sub
End Class

```

dataform1.aspx.vb kodları:

```

Imports System.Web.Security
Public Class DataWebForm1
    Inherits System.Web.UI.Page

    Dim cs, cv As DataView
    Dim ds As New DataSet
    Dim dr As DataRow
    Dim sira As String
    Dim adT As System.Web.UI.WebControls.TextBox
    Dim adTy As String
    Dim no_T As System.Web.UI.WebControls.TextBox
    Dim no_Ty As String
    Dim telT As System.Web.UI.WebControls.TextBox
    Dim tely As String
    Dim emailT As System.Web.UI.WebControls.TextBox
    Dim emaily As String
    Dim not1T As System.Web.UI.WebControls.TextBox
    Dim notly As String
    Dim not2T As System.Web.UI.WebControls.TextBox
    Dim eskiad As String
    Dim not2y As String
    Dim eskino As String
    Dim eskitel As String
    Dim eskiemail As String
    Dim eskinot1 As String
    Dim eskinot2 As String
    Protected WithEvents SqlSelectCommand1 As System.Data.SqlClient.SqlCommand
    Protected WithEvents SqlInsertCommand1 As System.Data.SqlClient.SqlCommand
    Protected WithEvents Label2 As System.Web.UI.WebControls.Label
    Protected WithEvents siraLabel As System.Web.UI.WebControls.Label
    Protected WithEvents yonLabel As System.Web.UI.WebControls.Label
    Protected WithEvents Label5 As System.Web.UI.WebControls.Label
    Protected WithEvents HyperLink1 As System.Web.UI.WebControls.HyperLink
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
    Protected WithEvents SqlCommand1 As System.Data.SqlClient.SqlCommand
    Protected WithEvents Label3 As System.Web.UI.WebControls.Label
    Protected WithEvents Button4 As System.Web.UI.WebControls.Button
    Protected WithEvents Button2 As System.Web.UI.WebControls.Button
    Protected WithEvents Button1 As System.Web.UI.WebControls.Button
    Protected WithEvents TextBox5 As System.Web.UI.WebControls.TextBox
    Protected WithEvents TextBox3 As System.Web.UI.WebControls.TextBox
    Protected WithEvents TextBox2 As System.Web.UI.WebControls.TextBox
    Protected WithEvents TextBox1 As System.Web.UI.WebControls.TextBox
    Protected WithEvents TextBox4 As System.Web.UI.WebControls.TextBox
    Protected WithEvents Button3 As System.Web.UI.WebControls.Button
    Protected WithEvents TextBox6 As System.Web.UI.WebControls.TextBox
    Protected WithEvents Button5 As System.Web.UI.WebControls.Button
    Protected WithEvents CheckBox1 As System.Web.UI.WebControls.CheckBox
    Protected WithEvents DataView1 As System.Data.DataView
    Protected WithEvents DataGrid1 As System.Web.UI.WebControls.DataGrid
    Protected WithEvents SqlCommand2 As System.Data.SqlClient.SqlCommand

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.objDataSet21 = New wa2.DataSet21
        Me.SqlConnection1 = New System.Data.SqlClient.SqlConnection
        Me.SqlDataAdapter1 = New System.Data.SqlClient.SqlDataAdapter
        Me.SqlCommand1 = New System.Data.SqlClient.SqlCommand
        Me.SqlInsertCommand1 = New System.Data.SqlClient.SqlCommand
        Me.SqlSelectCommand1 = New System.Data.SqlClient.SqlCommand
        Me.SqlCommand2 = New System.Data.SqlClient.SqlCommand
    End Sub

```

```

Me.DataView1 = New System.Data.DataView
CType(Me.objDataSet21, System.ComponentModel.ISupportInitialize).BeginInit()
CType(Me.DataView1, System.ComponentModel.ISupportInitialize).BeginInit()
'
'objDataSet21
'
Me.objDataSet21.DataSetName = "DataSet21"
Me.objDataSet21.Locale = New System.Globalization.CultureInfo("tr-TR")
'
'SqlConnection1
'
Me.SqlConnection1.ConnectionString = "workstation id=EV;packet size=4096;integrated security=SSPI;data
source=ev;persist" & _
"t security info=False;initial catalog=dtname"
'
'SqlDataAdapter1
'
Me.SqlDataAdapter1.DeleteCommand = Me.SqlCommand1
Me.SqlDataAdapter1.InsertCommand = Me.SqlInsertCommand1
Me.SqlDataAdapter1.SelectCommand = Me.SqlSelectCommand1
Me.SqlDataAdapter1.TableMappings.AddRange(New System.Data.Common.DataTableMapping() {New
System.Data.Common.DataTableMapping("Table", "tablo1", New System.Data.Common.DataColumnMapping() {New
System.Data.Common.DataColumnMapping("sira", "sira"), New System.Data.Common.DataColumnMapping("ad", "ad"), New
System.Data.Common.DataColumnMapping("no", "no"), New System.Data.Common.DataColumnMapping("tel", "tel"), New
System.Data.Common.DataColumnMapping("email", "email"), New System.Data.Common.DataColumnMapping("not1",
"not1"), New System.Data.Common.DataColumnMapping("not2", "not2")}})})
Me.SqlDataAdapter1.UpdateCommand = Me.SqlCommand2
'
'SqlCommand1
'
Me.SqlCommand1.CommandText = "DELETE FROM tablo1 WHERE (ad = @ad)"
Me.SqlCommand1.Connection = Me.SqlConnection1
Me.SqlCommand1.Parameters.Add(New System.Data.SqlClient.SqlParameter("@ad",
System.Data.SqlDbType.VarChar, 10, System.Data.ParameterDirection.Input, False, CType(0, Byte), CType(0, Byte),
"ad", System.Data.DataRowVersion.Original, Nothing))
'
'SqlInsertCommand1
'
Me.SqlInsertCommand1.CommandText = "INSERT INTO tablo1 (sira, ad, no, tel, email, not1, not2) VALUES
(@sira, @ad, @no" & _
", @tel, @email, @not1, @not2); SELECT sira, ad, no, tel, email, not1, not2 FROM " & _
"tablo1"
Me.SqlInsertCommand1.Connection = Me.SqlConnection1
Me.SqlInsertCommand1.Parameters.Add(New System.Data.SqlClient.SqlParameter("@sira",
System.Data.SqlDbType.VarChar, 10, "sira"))
Me.SqlInsertCommand1.Parameters.Add(New System.Data.SqlClient.SqlParameter("@ad",
System.Data.SqlDbType.VarChar, 10, "ad"))
Me.SqlInsertCommand1.Parameters.Add(New System.Data.SqlClient.SqlParameter("@no",
System.Data.SqlDbType.VarChar, 10, "no"))
Me.SqlInsertCommand1.Parameters.Add(New System.Data.SqlClient.SqlParameter("@tel",
System.Data.SqlDbType.VarChar, 10, "tel"))
Me.SqlInsertCommand1.Parameters.Add(New System.Data.SqlClient.SqlParameter("@email",
System.Data.SqlDbType.VarChar, 10, "email"))
Me.SqlInsertCommand1.Parameters.Add(New System.Data.SqlClient.SqlParameter("@not1",
System.Data.SqlDbType.Float, 8, "not1"))
Me.SqlInsertCommand1.Parameters.Add(New System.Data.SqlClient.SqlParameter("@not2",
System.Data.SqlDbType.Float, 8, "not2"))
'
'SqlSelectCommand1
'
Me.SqlSelectCommand1.CommandText = "SELECT sira, ad, no, tel, email, not1, not2 FROM tablo1"
Me.SqlSelectCommand1.Connection = Me.SqlConnection1
'
'SqlCommand2
'
Me.SqlCommand2.CommandText = "UPDATE tablo1 SET sira = @sira, ad = @ad, no = @no, tel = @tel, email =
@email, n" & _
"ot1 = @not1, not2 = @not2 WHERE (ad = @eskiad) AND (no = @eskino) AND (tel = @es" & _
"kitel) AND (email = @eskiemail) AND (not1 = @eskinot1) AND (not2 = @eskinot2)"
Me.SqlCommand2.Connection = Me.SqlConnection1
Me.SqlCommand2.Parameters.Add(New System.Data.SqlClient.SqlParameter("@sira",
System.Data.SqlDbType.VarChar, 10, "sira"))
Me.SqlCommand2.Parameters.Add(New System.Data.SqlClient.SqlParameter("@ad",
System.Data.SqlDbType.VarChar, 10, "ad"))

```

```

        Me.SqlCommand2.Parameters.Add(New System.Data.SqlClient.SqlParameter("@no",
System.Data.SqlDbType.VarChar, 10, "no"))
        Me.SqlCommand2.Parameters.Add(New System.Data.SqlClient.SqlParameter("@tel",
System.Data.SqlDbType.VarChar, 10, "tel"))
        Me.SqlCommand2.Parameters.Add(New System.Data.SqlClient.SqlParameter("@email",
System.Data.SqlDbType.VarChar, 10, "email"))
        Me.SqlCommand2.Parameters.Add(New System.Data.SqlClient.SqlParameter("@not1",
System.Data.SqlDbType.Float, 8, "not1"))
        Me.SqlCommand2.Parameters.Add(New System.Data.SqlClient.SqlParameter("@not2",
System.Data.SqlDbType.Float, 8, "not2"))
        Me.SqlCommand2.Parameters.Add(New System.Data.SqlClient.SqlParameter("@eskiad",
System.Data.SqlDbType.VarChar, 10, System.Data.ParameterDirection.Input, False, CType(0, Byte), CType(0, Byte),
"ad", System.Data.DataRowVersion.Original, Nothing))
        Me.SqlCommand2.Parameters.Add(New System.Data.SqlClient.SqlParameter("@eskino",
System.Data.SqlDbType.VarChar, 10, System.Data.ParameterDirection.Input, False, CType(0, Byte), CType(0, Byte),
"no", System.Data.DataRowVersion.Original, Nothing))
        Me.SqlCommand2.Parameters.Add(New System.Data.SqlClient.SqlParameter("@eskitel",
System.Data.SqlDbType.VarChar, 10, System.Data.ParameterDirection.Input, False, CType(0, Byte), CType(0, Byte),
"tel", System.Data.DataRowVersion.Original, Nothing))
        Me.SqlCommand2.Parameters.Add(New System.Data.SqlClient.SqlParameter("@eskiemail",
System.Data.SqlDbType.VarChar, 10, System.Data.ParameterDirection.Input, False, CType(0, Byte), CType(0, Byte),
"email", System.Data.DataRowVersion.Original, Nothing))
        Me.SqlCommand2.Parameters.Add(New System.Data.SqlClient.SqlParameter("@eskinot1",
System.Data.SqlDbType.Float, 8, System.Data.ParameterDirection.Input, False, CType(0, Byte), CType(0, Byte),
"not1", System.Data.DataRowVersion.Original, Nothing))
        Me.SqlCommand2.Parameters.Add(New System.Data.SqlClient.SqlParameter("@eskinot2",
System.Data.SqlDbType.Float, 8, System.Data.ParameterDirection.Input, False, CType(0, Byte), CType(0, Byte),
"not2", System.Data.DataRowVersion.Original, Nothing))
    '
    'DataView1
    '
    Me.DataView1.Table = Me.objDataSet21.tablo1
    CType(Me.objDataSet21, System.ComponentModel.ISupportInitialize).EndInit()
    CType(Me.DataView1, System.ComponentModel.ISupportInitialize).EndInit()

End Sub
Protected WithEvents objDataSet21 As wa2.DataSet21
Protected WithEvents SqlConnection1 As System.Data.SqlClient.SqlConnection
Protected WithEvents SqlDataAdapter1 As System.Data.SqlClient.SqlDataAdapter
Protected WithEvents tablo As System.Web.UI.WebControls.DataGrid

'NOTE: The following placeholder declaration is required by the Web Form Designer.
'Do not delete or move it.
Private designerPlaceholderDeclaration As System.Object

Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init
    'CODEGEN: This method call is required by the Web Form Designer
    'Do not modify it using the code editor.
    InitializeComponent()
    ds = objDataSet21
    SqlDataAdapter1.SelectCommand.CommandText = "select * from tablo1 order by ad asc"
    siraLabel.Text = "ad"
    yonLabel.Text = "asc"
    sira = "ad"
    Label2.Text = ""
    tablo.SelectedIndex = -1
End Sub

#End Region

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    'Put user code to initialize the page here

    If (Trim(Application("Ad")) <> "") And (Trim(Application("Sifre")) <> "") Then
        LoadDataSet()
    Else
        Response.Write("<script> alert ('$ifresiz Giriş Yapılamaz!')</script>")
        Button1.Enabled = False
        Button2.Enabled = False
        Button3.Enabled = False
        Response.Redirect("index.aspx")
    End If
    Dim a As String
    a = Application("Ad")
    Label3.Text = "<b>Kullanıcı:</b> " & a

```

```

If Request("filtrele") <> "" Then
    DataView1.RowFilter = "ad like '%" & Request("filtrele") & "%'"
    DataView1.Sort = siraLabel.Text & " " & yonLabel.Text
    DataGrid1.DataSource = DataView1
    DataGrid1.DataBind()
    DataGrid1.Visible = True
    Label1.Text = "<b>Filtre metni:</b> *" & Request("filtrele") & "*"
Else
    DataGrid1.Visible = False
End If
If Request("silinsin") = "evet" Then
    Label1.Text = "<b>Silinen kayıt:</b> " & Request("silinecek")
    SqlDataAdapter1.DeleteCommand.Connection.Open()
    SqlDataAdapter1.DeleteCommand.Parameters.Item(0).Value = Request("silinecek")
    SqlDataAdapter1.DeleteCommand.ExecuteNonQuery()
    SqlDataAdapter1.DeleteCommand.Connection.Close()
    Response.Write("<script> document.location.href='dataform1.aspx';</script>")
End If
'tablo.PagerStyle.NextPageText = "Sonraki - Aktif sayfa no: " & tablo.CurrentPageIndex
'yonLabel.Text = "DESC"
'yon = False

End Sub

Private Sub buttonLoad_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Try
        Me.LoadDataSet()
        Me.tablo.SelectedIndex = -1
        Me.tablo.DataBind()
    Catch eLoad As System.Exception
        Me.Response.Write(eLoad.Message)
    End Try
End Sub

Function bul2(ByVal ad As String)
    Dim sonuc As String = "" 'bulunamazsa boşluk olur
    Dim s As String = "data source=localhost;initial catalog=dbname;integrated security=true"
    Dim conn As New SqlConnection(s)
    Dim dp1 As New SqlParameter("@ad", SqlDbType.Char)
    Dim da As New SqlDataAdapter("select * from tablo1 where ad=@ad", conn)
    Dim ds As New DataSet
    dp1.Value = ad
    da.SelectCommand.Parameters.Add(dp1)
    conn.Open()
    da.Fill(ds, "tablo1")
    If ds.Tables(0).Rows.Count > 0 Then
        sonuc = ds.Tables(0).Rows(0)(1)
    End If
    conn.Close()
    Return sonuc
End Function

Public Sub LoadDataSet()
    'Create a new dataset to hold the records returned from the call to FillDataSet.
    'A temporary dataset is used because filling the existing dataset would
    'require the databindings to be rebound.
    sira = siraLabel.Text
    If CheckBox1.Checked Then
        yonLabel.Text = "ASC"
    Else
        yonLabel.Text = "DESC"
    End If
    If Request("filtrele") = "" Then DataGrid1.Visible = False
    SqlDataAdapter1.SelectCommand.CommandText = "select * from tablo1 order by " & sira & " " &
yonLabel.Text
    Dim objDataSetTemp As wa2.DataSet21
    objDataSetTemp = New wa2.DataSet21
    Try
        'Attempt to fill the temporary dataset.
        objDataSetTemp = CType(Me.objDataSet21.Clone, wa2.DataSet21)
    Catch eFillDataSet As System.Exception
        'Add your error handling code here.
        Throw eFillDataSet
    End Try
    Try
        'Empty the old records from the dataset.

```

```

        objDataSet21.Clear()
        'Merge the records into the main dataset.
        objDataSet21.Merge(objDataSetTemp)
    Catch eLoadMerge As System.Exception
        'Add your error handling code here.
        Throw eLoadMerge
    End Try
    SqlDataAdapter1.Fill(objDataSet21.tablo1)
    tablo.SelectedIndex = -1
    tablo.DataBind()
    DataGrid1.DataBind()
    Label2.Text = "<b>Sıralama: </b>" & sira & "<br><b>Yön:</b> " & _
        yonLabel.Text & "<br><b>İşlemler</b>: sıralama(çift yön), listeleme(10'arlı sayfalama),
silme(onaylı)," & _
        " ekleme, güncelleme, arama, filtreleme (ayrı bir tabloda), hesaplanan alanlar(ayrı tabloda),
kullanıcı girişi (kullanıcı ekleme, şifre hatırlatma)" & _
        "<br> <i>(*) tam bitmedi</i><br><b>Kayıt sayısı:</b>" & _
        Val(objDataSet21.tablo1.Count)
    'tablo.PagerStyle.NextPageText = "Sonraki - Aktif sayfa no: " & tablo.CurrentPageIndex

End Sub
Private Sub tablo_PageIndexChanged(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridPageChangedEventArgs) Handles tablo.PageIndexChanged
    tablo.CurrentPageIndex = e.NewPageIndex
    Label5.Text = tablo.CurrentPageIndex
    'Session("sira") = sira
    tablo.EditItemIndex = -1
    LoadDataSet()
End Sub

Private Sub tablo_EditCommand(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridCommandEventArgs) Handles tablo.EditCommand
    tablo.EditItemIndex = e.Item.ItemIndex
    tablo.SelectedIndex = e.Item.ItemIndex

    adTy = tablo.SelectedItem.Cells(1).Text
    no_Ty = tablo.SelectedItem.Cells(2).Text
    tely = tablo.SelectedItem.Cells(3).Text
    emaily = tablo.SelectedItem.Cells(4).Text
    notly = tablo.SelectedItem.Cells(5).Text
    not2y = tablo.SelectedItem.Cells(6).Text

    TextBox1.Text = IIf(adTy = "&nbsp;", "", adTy)
    TextBox2.Text = IIf(no_Ty = "&nbsp;", "", no_Ty)
    TextBox3.Text = IIf(tely = "&nbsp;", "", tely)
    TextBox4.Text = IIf(emaily = "&nbsp;", "", emaily)
    TextBox5.Text = IIf(notly = "&nbsp;", "", notly)
    TextBox6.Text = IIf(not2y = "&nbsp;", "", not2y)
    Button1.Enabled = True
    Button3.Enabled = True
    LoadDataSet()
End Sub

Private Sub tablo_DeleteCommand(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridCommandEventArgs) Handles tablo.DeleteCommand
    Label11.Text = "<b>Silinen kayıt:</b> " & tablo.Items(e.Item.ItemIndex).Cells(1).Text
    SqlDataAdapter1.DeleteCommand.Connection.Open()
    SqlDataAdapter1.DeleteCommand.Parameters.Item(0).Value = tablo.Items(e.Item.ItemIndex).Cells(1).Text
    SqlDataAdapter1.DeleteCommand.ExecuteNonQuery()
    SqlDataAdapter1.DeleteCommand.Connection.Close()
    tablo.EditItemIndex = -1
    LoadDataSet()
End Sub

Private Sub tablo_CancelCommand(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridCommandEventArgs) Handles tablo.CancelCommand
    tablo.EditItemIndex = -1
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""
    TextBox6.Text = ""
    Button1.Enabled = False
    Button3.Enabled = False

```

```

        LoadDataSet()
    End Sub
    'kullanılmıyor
    Private Sub tablo_UpdateCommand(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridCommandEventArgs) Handles tablo.UpdateCommand
        SqlDataAdapter1.UpdateCommand.Connection.Open()
        SqlDataAdapter1.UpdateCommand.Parameters.Item(0).Value = ""
        SqlDataAdapter1.UpdateCommand.Parameters.Item(1).Value = CType(e.Item.Cells(1).Controls(0),
System.Web.UI.WebControls.TextBox).Text
        SqlDataAdapter1.UpdateCommand.Parameters.Item(2).Value = CType(e.Item.Cells(2).Controls(0),
System.Web.UI.WebControls.TextBox).Text
        SqlDataAdapter1.UpdateCommand.Parameters.Item(3).Value = CType(e.Item.Cells(3).Controls(0),
System.Web.UI.WebControls.TextBox).Text
        SqlDataAdapter1.UpdateCommand.Parameters.Item(4).Value = CType(e.Item.Cells(4).Controls(0),
System.Web.UI.WebControls.TextBox).Text
        SqlDataAdapter1.UpdateCommand.Parameters.Item(5).Value = Val(CType(e.Item.Cells(5).Controls(0),
System.Web.UI.WebControls.TextBox).Text)
        SqlDataAdapter1.UpdateCommand.Parameters.Item(6).Value = Val(CType(e.Item.Cells(6).Controls(0),
System.Web.UI.WebControls.TextBox).Text)
        SqlDataAdapter1.UpdateCommand.Parameters.Item(7).Value = e.Item.Cells(1).Text
        SqlDataAdapter1.UpdateCommand.ExecuteNonQuery()
        SqlDataAdapter1.UpdateCommand.Connection.Close()

        Label1.Text = "Kaydedilen: " & CType(e.Item.Cells(1).Controls(0),
System.Web.UI.WebControls.TextBox).Text & ", eski ad=" & e.Item.Cells(1).Text

        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
        eskiad = ""
        Button1.Enabled = False
        Button3.Enabled = False
        tablo.EditItemIndex = -1
        LoadDataSet()
    End Sub
    'siralama
    Private Sub tablo_SortCommand(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridSortCommandEventArgs) Handles tablo.SortCommand
        Dim cv As DataView
        cv = New DataView(ds.Tables("tablo1"))
        cv.Sort = e.SortExpression

        Label1.Text = "<b>Sıralanan alan:</b> " & cv.Sort
        sira = e.SortExpression
        siraLabel.Text = sira
        tablo.EditItemIndex = -1
        LoadDataSet()
    End Sub
    'seç komutu
    Private Sub tablo_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
tablo.SelectedIndexChanged
        Label1.Text = "<b>Seçili Kayıt:</b> " & tablo.SelectedItem.Cells(1).Text
        adTy = tablo.SelectedItem.Cells(1).Text
        no_Ty = tablo.SelectedItem.Cells(2).Text
        tely = tablo.SelectedItem.Cells(3).Text
        emaily = tablo.SelectedItem.Cells(4).Text
        notly = tablo.SelectedItem.Cells(5).Text
        not2y = tablo.SelectedItem.Cells(6).Text
        If eskiad = "" Then eskiad = tablo.SelectedItem.Cells(1).Text
        TextBox1.Text = IIf(adTy = "&nbsp;", "", adTy)
        TextBox2.Text = IIf(no_Ty = "&nbsp;", "", no_Ty)
        TextBox3.Text = IIf(tely = "&nbsp;", "", tely)
        TextBox4.Text = IIf(emaily = "&nbsp;", "", emaily)
        TextBox5.Text = IIf(notly = "&nbsp;", "", notly)
        TextBox6.Text = IIf(not2y = "&nbsp;", "", not2y)
        Button1.Enabled = True
        Button3.Enabled = True
        Session.Add("eskiad", adTy)
        Session.Add("eskino", no_Ty)
        Session.Add("eskitel", tely)
        Session.Add("eskiemail", emaily)
        Session.Add("eskinot1", notly)

```

```

        Session.Add("eskinot2", not2y)
    End Sub
    'kullanılmıyor
    Private Sub tablo_ItemDataBound(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.DataGridItemEventArgs) Handles tablo.ItemDataBound
        Dim i As ListItemType
        i = CType(e.Item.ItemType, ListItemType)
        'If sira <> "" Then CType(e.Item.Controls(sira + 2), TableCell).Font.Bold = True
    End Sub

    'kaydet
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        If Trim(TextBox1.Text) <> "" Then
            eskiad = Session("eskiad")
            eskino = Session("eskino")
            eskitel = Session("eskitel")
            eskiemail = Session("eskiemail")
            eskinot1 = Session("eskinot1")
            eskinot2 = Session("eskinot2")
            adTy = TextBox1.Text
            no_Ty = TextBox2.Text
            tely = TextBox3.Text
            emaily = TextBox4.Text
            notly = TextBox5.Text
            not2y = TextBox6.Text

            SqlDataAdapter1.UpdateCommand.Connection.Open()
            SqlDataAdapter1.UpdateCommand.Parameters.Item(0).Value = ""
            SqlDataAdapter1.UpdateCommand.Parameters.Item(1).Value = adTy
            SqlDataAdapter1.UpdateCommand.Parameters.Item(2).Value = no_Ty
            SqlDataAdapter1.UpdateCommand.Parameters.Item(3).Value = tely
            SqlDataAdapter1.UpdateCommand.Parameters.Item(4).Value = emaily
            SqlDataAdapter1.UpdateCommand.Parameters.Item(5).Value = Val(notly)
            SqlDataAdapter1.UpdateCommand.Parameters.Item(6).Value = Val(not2y)
            SqlDataAdapter1.UpdateCommand.Parameters.Item(7).Value = eskiad
            SqlDataAdapter1.UpdateCommand.Parameters.Item(8).Value = eskino
            SqlDataAdapter1.UpdateCommand.Parameters.Item(9).Value = eskitel
            SqlDataAdapter1.UpdateCommand.Parameters.Item(10).Value = eskiemail
            SqlDataAdapter1.UpdateCommand.Parameters.Item(11).Value = eskinot1
            SqlDataAdapter1.UpdateCommand.Parameters.Item(12).Value = eskinot2
            SqlDataAdapter1.UpdateCommand.ExecuteNonQuery()
            SqlDataAdapter1.UpdateCommand.Connection.Close()

            Label1.Text = "Kaydedilen: " & adTy & ", eski ad=" & eskiad

            TextBox1.Text = ""
            TextBox2.Text = ""
            TextBox3.Text = ""
            TextBox4.Text = ""
            TextBox5.Text = ""
            TextBox6.Text = ""
            eskiad = ""
            Button1.Enabled = False
            Button3.Enabled = False
        Else
            Label1.Text = "Kayıt tamamlanamadı! Boş ad girişi yapılamaz."
        End If
        LoadDataSet()
    End Sub
    'Ekle
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        Dim varmi As String
        varmi = bul2(TextBox1.Text)
        If varmi = "" And Trim(TextBox1.Text) <> "" Then
            adTy = TextBox1.Text
            no_Ty = TextBox2.Text
            tely = TextBox3.Text
            emaily = TextBox4.Text
            notly = TextBox5.Text
            not2y = TextBox6.Text

            SqlDataAdapter1.InsertCommand.Connection.Open()
            SqlDataAdapter1.InsertCommand.Parameters.Item(0).Value = ""
            SqlDataAdapter1.InsertCommand.Parameters.Item(1).Value = adTy
            SqlDataAdapter1.InsertCommand.Parameters.Item(2).Value = no_Ty

```

```

        SqlDataAdapter1.InsertCommand.Parameters.Item(3).Value = tely
        SqlDataAdapter1.InsertCommand.Parameters.Item(4).Value = emaily
        SqlDataAdapter1.InsertCommand.Parameters.Item(5).Value = Val(not1y)
        SqlDataAdapter1.InsertCommand.Parameters.Item(6).Value = Val(not2y)
        SqlDataAdapter1.InsertCommand.ExecuteNonQuery()
        SqlDataAdapter1.InsertCommand.Connection.Close()

        Label1.Text = "Eklenen: " & SqlDataAdapter1.InsertCommand.Parameters.Item(1).Value

        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
    Else
        Label1.Text = "Kayıt tamamlanamadı! Ad boş olamaz veya aynı ad ile giriş yapılamaz. " &
        TextBox1.Text
    End If
    LoadDataSet()
End Sub

Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
    If TextBox1.Text <> "" Then Button1.Enabled = True Else Button1.Enabled = False
    If TextBox1.Text <> "" Then Button3.Enabled = True Else Button3.Enabled = False
End Sub
'sil
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    Response.Write("<script> " & _
    "cev=confirm('Seçili kayıt \" & TextBox1.Text & "\" silinsin mi?'); " & _
    "if (cev) document.location.href='dataform1.aspx?silinecek=evet&silinecek=" & TextBox1.Text & "'; " & _
    "</script>")
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""
    TextBox6.Text = ""
    Label1.Text = "<b>Seçim iptal edildi...</b>"
    Button1.Enabled = False
    Button3.Enabled = False
    tablo.EditItemIndex = -1
    DataGrid1.Visible = False
    LoadDataSet()
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click
    Response.Write("<script> " & _
    "cev=prompt('Arama metni giriniz',''); " & _
    "if (cev) {cev='dataform1.aspx?filtrele='+cev;document.location.href=cev; }" & _
    "</script>")
End Sub

Private Sub DataGrid1_PageIndexChanged(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridPageChangedEventArgs) Handles DataGrid1.PageIndexChanged
    DataGrid1.CurrentPageIndex = e.NewPageIndex
End Sub
End Class

```


19. okul3d (Sanal Gerçeklik Örneği - VRML)

(Ornekler\Diger\okul3d4.zip)

Bu uygulamamızda, nihayet internet üzerinde de rahatlıkla çalışan bir 3d sayfamız oluyor. Yapım tarihinin sırasına göre aşağıda birkaç resmi görüyorsunuz.



Resim 23a – Yapım aşamalarına göre VRML site tasarımı

ISB – Internet Scene Builder (parallelgraphics.com) programı ile bu örnek yapıldı. Daha bitmedi aslında. Eksikler animasyon ve gerekli sağlam linkler. Animasyon ile ilgili en güzel çalışmam "Ornekler\Diger\VRMLcalismalarim.zip" içindeki evim.wrl. Kapılar otomatik açılabilir, nesneler sürüklenebilir ve şöminede ateş yanma efekti bile var. Biz de okul3d'de arabaları hareket ettirebiliriz, kapıları sürgülü yapabiliriz, açılabilen raflar, kitaplık gibi şeyler ekleyebiliriz. outline3d.com sitesinde güzel emlakçılık örnekleri var. Üye olarak yaptığınız tasarımı kaydedebilirsiniz. Çok aramama rağmen sohbet kodlarını bulamadım. 3d sohbet çok ilginç olabilirdi.

VRML'nin bilgisayarımızda çalışması için örneğin Internet Explorer için Cortona, CosmoPlayer veya Blaxxun Contact gibi eklentiler gerekli. Birini rahatlıkla internet üzerinden indirebilirsiniz. Örneğin Cortona'yı yüklediğimizde artık IE *.wrl dosyaları tanıyor ve açabiliyor. Resimlerde de görüldüğü gibi sol ve alt kısımlarda araç çubukları ile yardımcı olmaya çalışıyor. Sahneye sağ tıklayıp diğer ayarlara ulaşabilirsiniz. Seçeneklerde dikkatimi çeken bir ayar da

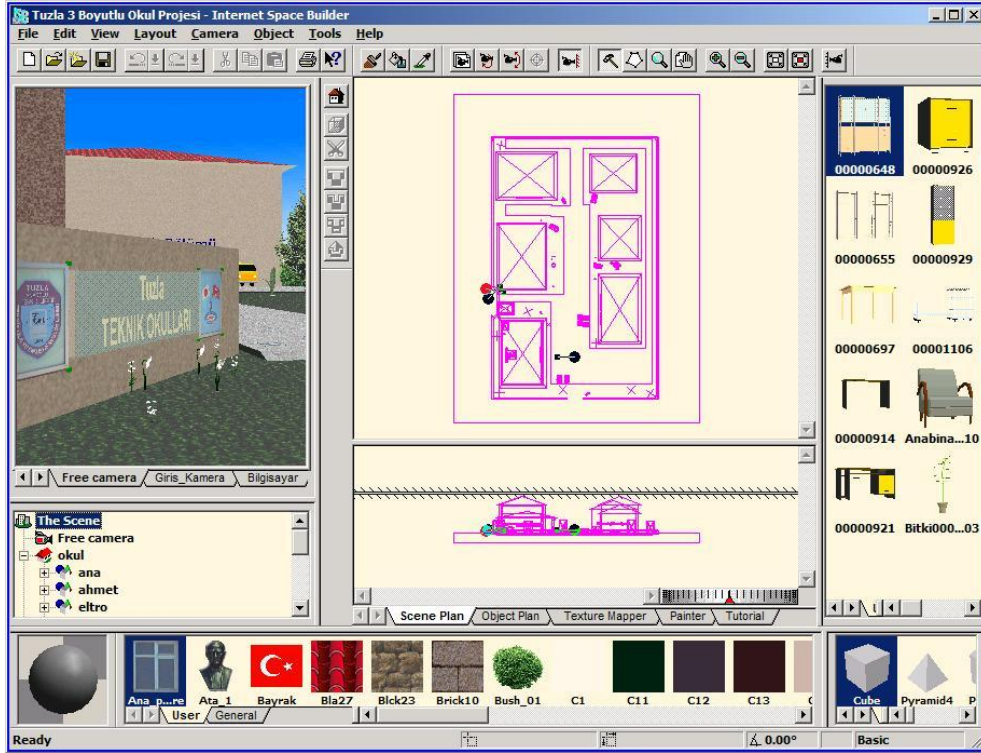
DirectX, OpenGL ve kendi 3d motoru arasında rahatlıkla geiş yapılabilmesi. FPS – frame per second hızınızı artırmak için ayarlar ile oynayabilirsiniz. Ya da makineni iyi ise örneğın real-time – antialiasing (gerçek zamanlı kenar düzeltme) yapabilirsiniz.

Sahnedeki bazı resimleri, dijital fotoğraf makinemle çektim. Mesela duvardaki telefon gibi, yerdeki taşlar gibi. Ama abartıp daha gerçek gibi gözüksün diye heryeri bütün olarak çekip, duvarları öyle kaplamak zor bir iş. Oyunlarda genelde küçük parçalar halinde gerçek pencere, duvar, nesne(halı, tablo,...) resimleri çekilerek döşeniyor.

Animasyon için en kullanışlı program ISA – Internet Scene Assembler. Zaman çizgisi üzerinde, Flash programı gibi tasarımlar yapılıyor. Hiçbir programlama veya kod yazımı olmadan animasyon hazırlanabiliyor. Karakter animasyonu için kullanılabilecek program da ICA – Internet Character Animator.

Metin editörü olarak VRMLPad kullanılarak *.wrl dosyalar düz yazı dosyası olarak açılıp değiştirilebilir. Normalde çok uzun dosyalar *.zip gibi sıkıştırılarak IE içinde gösterilebiliyor. Sıkıştırmaya rağmen VRMLPad güzelce o dosyayı açıp düzenleyebiliyor.

Aklıma BlueDream'daki veya Yunus'da 3d çıkış – export olarak VRML de eklemek geliyor, ama çok uzun iş olduğu için şimdilik hayal olarak kalıyor. Aynı PHP ve ASP çıkış fikri gibi. Yunus'da HTML ile uğraşırken bile çok zaman harcamıştım.



Resim 23b – ISB tasarım ekranı

VRML tasarım aracı olarak bence en kullanışlı olan ISB'yi tavsiye ediyorum. 3DMax ile de VRML çıkış verilebiliyor. Ya da daha doğrusu tüm popüler tasarım araçları VRML desteklidir. Resim 23b'de görüldüğü gibi tek pencerede hemen canlı olarak hem tasarımı hem de çalışma halini görebiliyoruz. Aslında bu program VRML1.0 destekli. Yani dışarıdan nesne eklerken VRML2.0 ile yapılmış olanlarda sorun çıkıyor. Bir nesne veya sahneyi yüzde yüz aynısını yapacağım dersanız tek yapmanız gereken sabırla uğraşmak. İlle de tıpatıp aynısı olmak zorunda olmasa da, daha önce gidilmiş bir yeri internet üzerinde görmek ilginç. Bazı havaalanları, müzeler, şehirleri internet üzerinde bulabilirsiniz. Hatta bazı siteler turistik gezi bile yaptırıyorlar.

20. Net Pazar (PHP ve MySQL ile internette bit pazarı kurmak)

(Ornekler\Diger\netpazar.zip)

Bu projede yine PHP ile etkileşimli sayfalar yapıyoruz. 30 saatlik kısa bir çalışma ile iki hafta gibi sürede, 1650 satır koda yazıp, son haline geldik. Amacımız, insanların *üyelik* gerektirmeden (üyelik formları hep hatalı ve yanlış bilgilerle dolduruluyor ve şifre genelde unutuluyor) cep telefonu, kitap, otomobil gibi eşyalarını duyuru olarak bırakmaları, istendiğinde de duyurunun süresini uzatabilmeleridir. Duyuru yöneticinin izin vermesi ile aktif hale geliyor. Kabul tarihinden itibaren, eğer 7 günlük süre geçmişse, duyuru resmi de varsa kendini siliyor. Önceki projelerdeki gibi, burada da resim yollama ve eposta atma seçenekleri de var.

Kullandığım program bu defa Crimson Editor değil, Dreamweaver MX 2004 oldu. Kod tamamlama, renklendirme ve benzeri güzel özelliklerinden dolayı bu sefer DW'ı seçtim. Çok az resim var ama, yine de Fireworks ile bir iki düzeltme de yaptığımı söylemem gerek. DHTML kullanarak resimli duyuruları açmadan, fare ile üzerine gelince resimlerini görebiliyoruz. Layer (katman) nesnesinin görünür veya gizli olma halini yine JavaScript olmasa yapamayız. Veritabanını tablo olarak gösterirken yine çok gerekli olan sıralama, grup ya da blok halinde kayıtları listeleme (SQL'deki limit komutu tam bu iş içindir) ve arama da çok gerekli kodlamalardır.

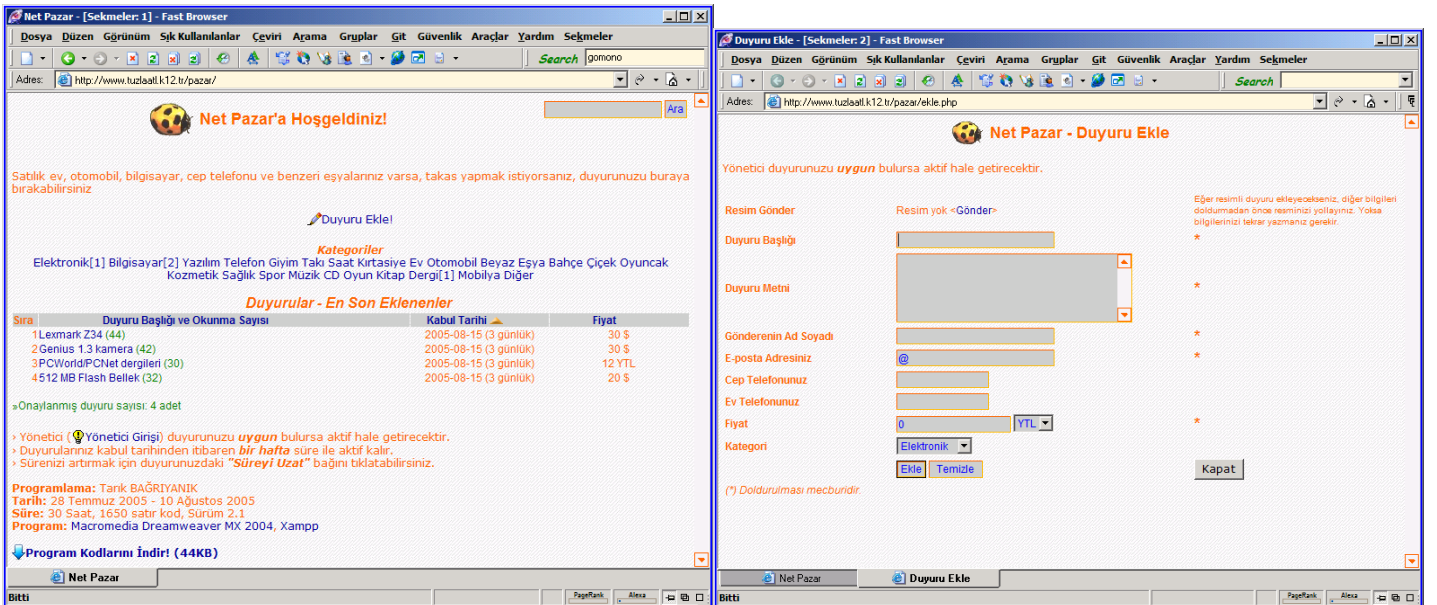
Kurulum adımları:

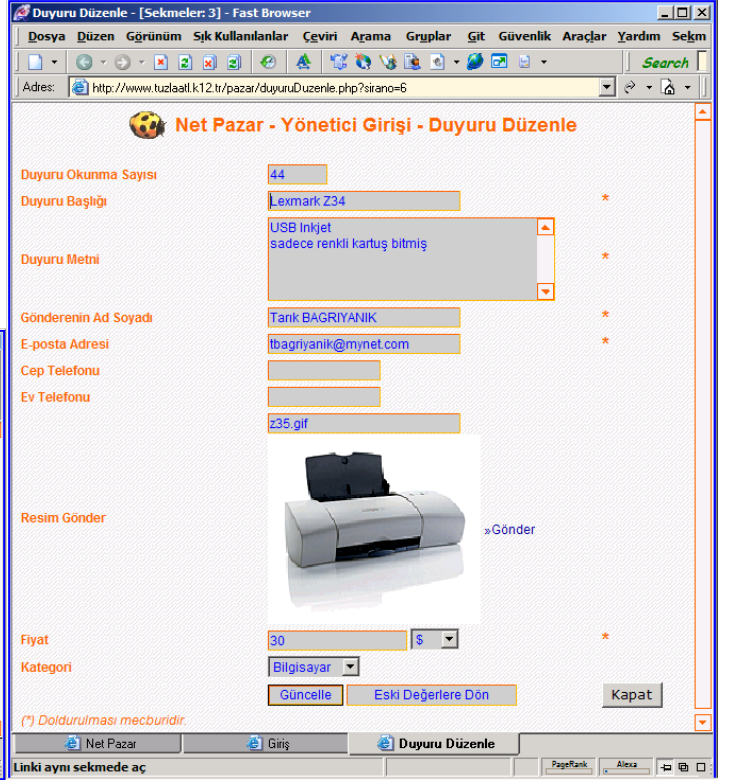
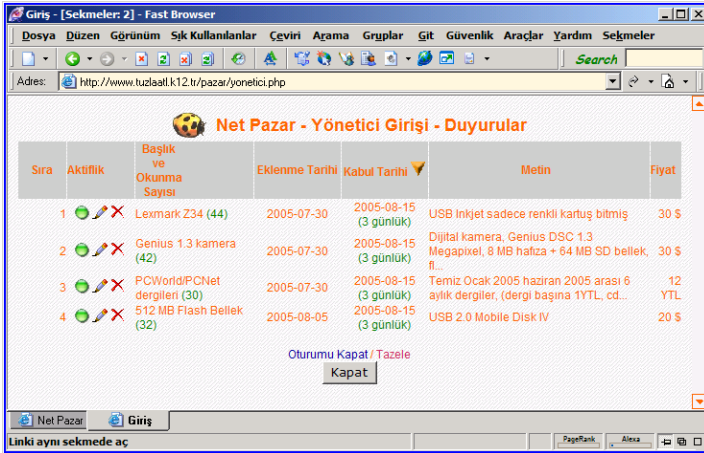
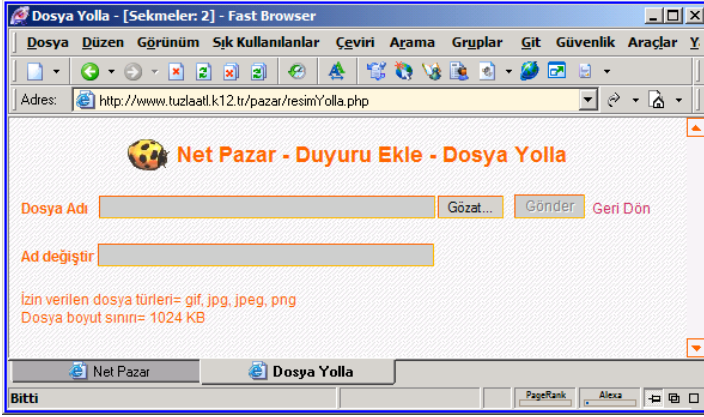
- 1- "apache/htdocs/pazar" dizinine dosyaları açınız.
- 2- phpMyAdmin arabiriminde "vt.sql" dosyasını çalıştırarak, 2 tabloyu (admin, netpazar) oluşturunuz.
- 3- Linux'da resim gönderimi için "apache/htdocs/pazar/yollanan" klasörüne yazma hakkı verin. Windows için buna gerek yoktur.
- 4- "conf.php" dosyasında gerekli ayarları yapınız.
- 5- "<http://localhost/pazar>" çalışmalıdır. Yöneticinin varsayılan adı:admin, şifresi:123'dür.

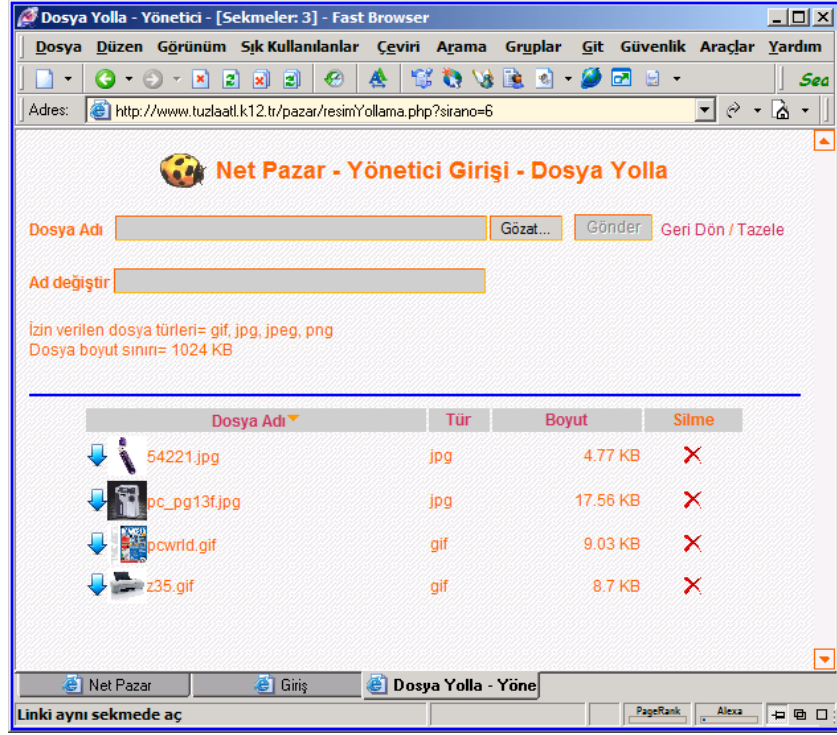
Duyuru ekleme fikri yerine, dosya, resim, program ekleme de rahatlıkla bu projede olabilir. Ya da tek başına download (dosya indirme) sitesi de yapılabilir. Eğer programcılıkla ilgili bir site yapılacaksa, kod yollama da olabilir. Kategoriler mesela C, Java, Basic gibi olabilir. Buna benzer ASP&MDB örneğimiz Bölüm 7 – 12. konudaki myasp'dir. 2001 tarihli çalışmamda brinkster.com'da çalışacak şekilde 3 modül; misafir defteri, link ekleme ve program ekleme sitesi tasarladım. PHP kadar gelişkin olmasa da...

Daha önce de dediğim gibi, internet programcılığı artık çok kolaylaşıyor. Kısa sürede güçlü veritabanı alt yapısı ile fonksiyonel siteler hazırlanabilir. Güvenlik sorunu da halledilince ticari uygulamalar rahatlıkla internette çalışacak şekilde yapılabilir.

Aslında ebay.com gibi duyurular üzerinde pazarlık, açık artırma da yapılabilir. Bunun için üyelik sistemi geliştirilmesi lazımdır. Hatta kredi kartı ile satın alma, hemen sipariş etme, paypal.com güvenli ağ aracılığı ile alışveriş de olabilir.



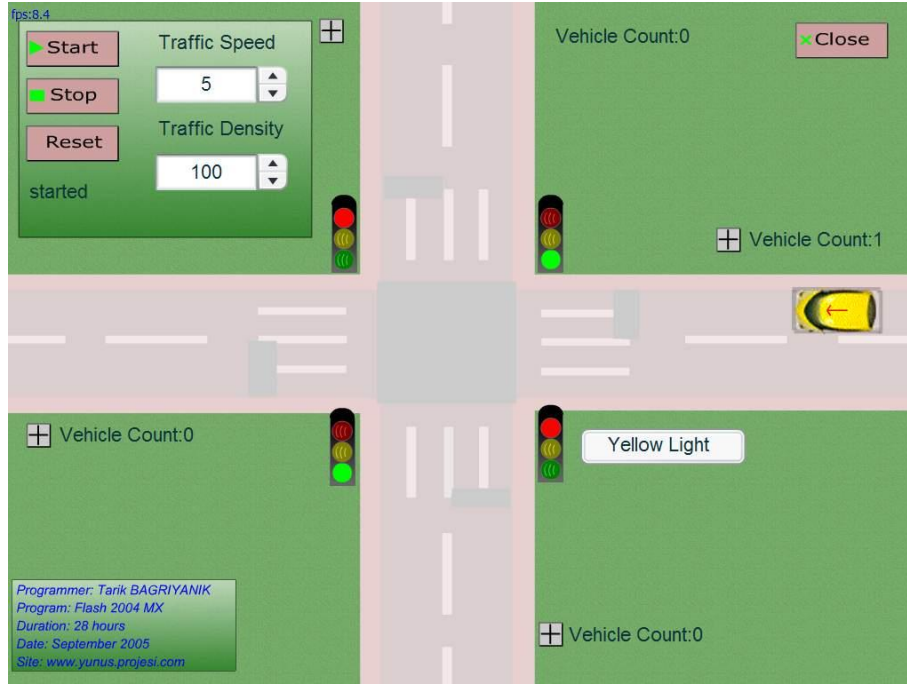




Resim 24 – Net Pazar’ın tüm ekran görüntüleri

21. Trafik Simulasyonu (Flash Action Script)

(Ornekler\Diger\trafik.zip)



Resim 25 – Trafik ekran görüntüsü

Flash MX 2004 ile daha doğrusu Action Script 2.0 ile bu projeyi tasarlıyoruz, programlıyoruz. Delphi kullanarak Beyoğlu Teknik Lisesinde buna benzer temrin yapmıştık. Ama kodları alamamıştım. Hemen hemen bu görüntüye sahipti. Bu proje üzerinde 1 haftacık uğraştım. Script kısmı biraz zorlarsa da animasyon için en iyi araç Flash galiba. VRML olarak da ISA programında 3 boyutlu kavşak örneği bulabilirsiniz.

Sarı ışık seçeneği ile arabaların serbest şekilde, önce gelen geçerek trafik akıyor. Normal ışıklara gelince araçların zamanla yığıldığı görülebilir. Birbirlerine çarpmamaları için etraflarında güvenlik şeridi var. Fakat bazen tam üst üste geldiklerinde iki araç da oldukları yerde kalıyor. Bunun dışında büyük bir hata yok bence. Trafik ışıklarının süresi çok kısa gibi. Aslında dinamik olarak, trafiğin yoğunluğuna göre, süre de değiştirilebilirdi. Temel olarak trafik ışıkları yayalar için yapılmış. Şimdilik onlar yok, diğer türde araçlar da yok. Yani kamyon, otobüs veya bisiklet de araç olarak eklenebilmelidir.

Araç sayısı arttıkça üst sol köşedeki fps (frame per second – saniyedeki kare sayısı) azalıyor. Aynen MyBoard projemde de zamanla yavaşlama oluyor hatırlarsanız. Normalde Delphi, VB veya başka dilde birçok iç içe for, while döngüsü rahatlıkla, hızı etkilemeden kullanılabilirken, Flash içinde hemen hız düşüyor. Belki kötü programla yöntemimden kaynaklanıyor veya her küçük filme (movieClip) kod eklenmesi sonsuz döngülere sebep oluyordur. Demek ki çok karışık olmayan, mesela fizik konusundaki kaldırıcı, kimya konusundaki mikroskop gibi ufak boyutlu animasyonları Flash ile yapmak, sonra da onları HTML, PHP, ASP ile birleştirmek daha uygun.

Daha önceki Flash ile kayan yazı örneğimizde de HTML ile Flash haberleşiyordu. Trafik ve MyBoard'ın Export ayarlarını *.exe yapmaya göre ayarladım. Ama bu iki proje web sayfası içinde de olabilirdi.

Buna benzer tren durakları simülasyonu da yapabiliriz. Daha karışık metro yolları, iç içe, alt geçitli, üst geçitli çözümlere de görsel olarak yardımcı olabiliriz.

Flash ile ilgili birkaç ipucu:

- 1) Tam ekran uygulama

```

$$fullscreen = true;
fscommand("fullscreen", "true");
_quality = "HIGH";

```

2) Tüm nesneleri tarama

```
for (name in _root)
{
    var obj = _root[name];
    var ad = obj._name;
    if (ad.substring(0, 9) == "bati_car_")
    {
        a++;
    }
}
```

3) Yeni nesne türetme

```
ara = _root.getNextHighestDepth();
_root.guney_araba.duplicateMovieClip("guney_car_"+ara, ara);
_root["guney_car_"+ara]._y = 200;
```

4) Zamanlayıcı(timer) oluşturma

```
function ilerle(){
    trace ("OK"); //output ekranına yazı yollama (Ctrl+Enter için geçerli)
}
myGlobalTimerlerle = setInterval(ilerle, 100);

if (_root.islemBasladi == false)
{
    clearInterval(myGlobalTimerlerle); //timer iptal edilir
}
```

5) Sahne yüklendiğinde dışarıdan film yüklenmesi

```
ifFrameLoaded (1)
{
    for (name in _root)
    {
        removeMovieClip(_root[name]); //nesne silme
    }
    loadMovieNum("timer.swf", 10); //aktif swf'nin hemen yanında bulunmalıdır.
}
```

6) Fare ipuçları yapma

```
on (rollOver) {
    this.araba_adi.text = this._name;
    this.araba_adi._visible = true;
}
on (rollOut) {
    this.araba_adi._visible = false;
}
```

7) Alt nesnelere nasıl ulaşırız?

```
for (name in _root)
{
    //tüm tiptouch lar gizlenir
    if (name.substring(0, 4) == "mlzm")
    {
        var objt = _root[name];
        for (each in objt) //işte burası!
        {
            var objin = objt[each];
            if (objin._name.substring(0, 8) == "TipTouch")
            {
                objin._visible = false;
            }
        }
    }
}
```

8) İki nokta arasındaki açının derecesi


```
aci = Math.atan((_root.sonNoktaY-_root.ilkNoktaY)/
  (_root.sonNoktaX-_root.ilkNoktaX))
*180/Math.PI;
```

9) Dışarıdan metin dosya ekleme
loadVariables("english.txt", "_root");

örnek içerik:

&durdur_metni=Stop

&calistir_metni=Run

...

10) Nesne seviyesini değiştirme

```
mx.behaviors.DepthControl.sendToBack(this);
```

```
mx.behaviors.DepthControl.bringForward(this);
```

11) İnternet adresi açmak

```
getURL("readme.txt","_blank");
```

12) Programı kapatmak

```
on (release) {
```

```
    fscommand("quit");
```

```
}
```

13) Rasgele komutu ve case komutu örneği

```
function yonBelirle(kim) {
```

```
    var objt = _root[kim];
```

```
    yon = random(3);
```

```
    switch (yon)
```

```
    {
```

```
    case 0 :
```

```
        objt.yon_ileri._rotation = 0;
```

```
        break;
```

```
    case 1 :
```

```
        objt.yon_ileri._rotation = 90;
```

```
        break;
```

```
    case 2 :
```

```
        objt.yon_ileri._rotation = 270;
```

```
        break;
```

```
    }
```

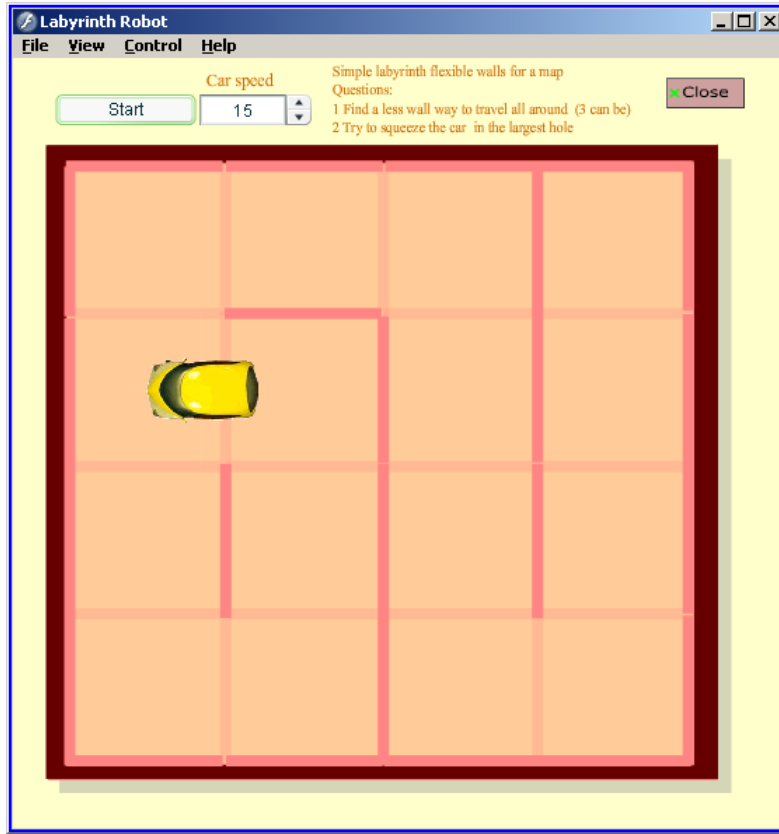
```
}
```


22. Labirent Robotu Simulasyonu (Flash 8 Action Script)

(Ornekler\Diger\labir4.zip)

4x4 hücrede bir robotun kendiliğinden yol bulması. Çok az kod yazarak bu hale geldi. Aslında gerçek bir yapay zeka programı değil. Esnek duvarların engellemesi ile araba kendi yolunu buluyor. Genelde hep belli bir yerde dönüp duruyor. Sadece sağ ve sol boş olunca random (rasgele sağ veya sol) karar veriyor. Okulda PIC ile yapılmış bir labirent robotundan örnek aldım. Resimde de o robotun yolları var. Benzer olarak çalıştığını göremedim ama, uzaktan kumandalı (RF radio frequency – radyo sinyali) bir başka proje vardı. Delphi ile paralel porttan sinyaller araca aktarılıyor ve ileri geri sağ sol düğmeleri ile araç hücrelerde geziyordu.

Daha geliştirilip, hedef hücreye en kısa hangi yoldan gidilir, harita kaydı, daha büyük alan, birçok araç, durak noktaları gibi konular eklenebilir. Araca dönme animasyonu da eklemedim. İlk bilgisayar oyunlarında da olan labirentte düşman ya da canavar sizi takip etmesi, tam tersi sizden kaçması algoritmalar ile yapılabilir. Bu tür kısa yoldan bir yere gitme problemi genelde oyunlar için yazılsa da, gerçek hayatta da telefon kabloları, cep telefonları ağları, ağ mimarilerinde de az maliyetli kurulum için gereklidir.



Resim 26 – Labirent ekran görüntüsü

Kodlar:

```
var arabaGit:Boolean = false;
var hiz:Number = 100;
_root.hizStepper.value=hiz;
$$fullscreen = true;
fscommand("fullscreen", "true");
_quality = "HIGH";
ifFrameLoaded (1) {
    _root.araba._rotation = 270;
}
function alphaAyar(gelenNesne) {
```

```

        if (gelenNesne._alpha == 100) {
            gelenNesne._alpha = 25;
        } else {
            gelenNesne._alpha = 100;
        }
    }
    //sensorler
    function onSensorDegdiMi() {
        sonuc = false;
        for (name in _root) {
            var obj = _root[name];
            var ad = obj._name;
            if (name.substring(0, 5) == "duvar" && obj._alpha != 25) {
                if (obj.hitTest(_root.araba.onSensor)) {
                    _root.araba.onSensor._alpha = 100;
                    sonuc = true;
                    break;
                }
            }
        }
        return sonuc;
    }
    function solSensorDegdiMi() {
        sonuc = false;
        for (name in _root) {
            var obj = _root[name];
            var ad = obj._name;
            if (name.substring(0, 5) == "duvar" && obj._alpha != 25) {
                if (obj.hitTest(_root.araba.solSensor)) {
                    _root.araba.solSensor._alpha = 100;
                    sonuc = true;
                    break;
                }
            }
        }
        return sonuc;
    }
    function sagSensorDegdiMi() {
        sonuc = false;
        for (name in _root) {
            var obj = _root[name];
            var ad = obj._name;
            if (name.substring(0, 5) == "duvar" && obj._alpha != 25) {
                if (obj.hitTest(_root.araba.sagSensor)) {
                    _root.araba.sagSensor._alpha = 100;
                    sonuc = true;
                    break;
                }
            }
        }
        return sonuc;
    }
    //action start or stop
    function ilerle() {
        _root.araba.onSensor._alpha = 10;
        _root.araba.solSensor._alpha = 10;
        _root.araba.sagSensor._alpha = 10;
        if (_root.arabaGit == true) {
            onSensorDegeri = onSensorDegdiMi();
            solSensorDegeri = solSensorDegdiMi();
            sagSensorDegeri = sagSensorDegdiMi();
            trace(onSensorDegeri + "-." + solSensorDegeri + "-." + sagSensorDegeri + ":" + _root.araba._rotation);
            //true - deđiyor, false - deđmiyor anlamındadır
            if (onSensorDegeri == true && sagSensorDegeri == true) {
                _root.araba._rotation -= 90;
                //sola dön - tek yol _|
            }
        }
    }

```

```

} else if (onSensorDegeri == true && solSensorDegeri == true) {
    _root.araba._rotation += 90;
    //sağa dön - tek yol -|
} else if (onSensorDegeri == true && solSensorDegeri == false && sagSensorDegeri == false) {
    yonNere = random(2);
    //IKI TARAF BOŞ
    if (yonNere == 0) {
        _root.araba._rotation += 90;
    }
    if (yonNere == 1) {
        _root.araba._rotation -= 90;
    }
    //sağa dön - tek yol -|
} else if (onSensorDegeri == true && solSensorDegeri == true && sagSensorDegeri == true) {
    _root.araba._rotation -= 180;
    //her yer dolu geri dön - tek yol =|
} else if (onSensorDegeri == false) {
    //ön boş
    trace(_root.araba._rotation);
    if (_root.araba._rotation == 90) {
        //yön kuzey
        _root.araba._y = _root.araba._y - 5;
    } else if (_root.araba._rotation == -90) {
        //yön güney
        _root.araba._y = _root.araba._y + 5;
    } else if (_root.araba._rotation == 0) {
        //yön doğu
        _root.araba._x = _root.araba._x - 5;
    } else if (_root.araba._rotation == -180 || _root.araba._rotation == 180) {
        //yön batı
        _root.araba._x = _root.araba._x + 5;
    }
}
}
}
myTimerllerle = setInterval(ilerle, _root.hiz);

```

23. dbase uygulamaları

(Ornekler\Diger\ogr kay_dbase.zip ve MEDIA_dbase.zip)

DOS tabanlı dbase programları ve veritabanları Visual Studio 6'nın bir elemanı olan Microsoft Visual FoxPro veya Visual dbase ile rahatlı açılabilir. 1990'lı yıllara geri döndük. Kod kısmının basit bir dil olduğunu ogr kay örneğinin ana programı ile görelim:

* program ANA MENU

set talk off

set echo off

set status off

set conf off

use ogr kay.dbf

do while .t.

clear

@1,10 say "Genel Öğrenci Değerlendirmesi"

@2,10 say "*****"

@5,10 say "1. Öğrenci not kaydı"

@7,10 say "2. Sorgulama"

@9,10 say "3. Yazıcı"

@11,10 say "4. Indexleme"

@13,10 say "9. Çıkış"

@18,10 say "Seçiminiz="

sec=" "

@18,40 get sec pict "!"

read

do case

case sec="1"

do og kay

loop

case sec="2"

do ogsor

loop

case sec="3"

do yaz

loop

case sec="4"

close all

use ogr kay

index on nosu tag no

loop

case sec="9"

clear

return

otherwise

loop

endcase

enddo

return

ekran ayarlarını yapıyoruz

veritabanını seç

sonsuz döngü

ekran temizle

menü

klavyeden giriş biçimi

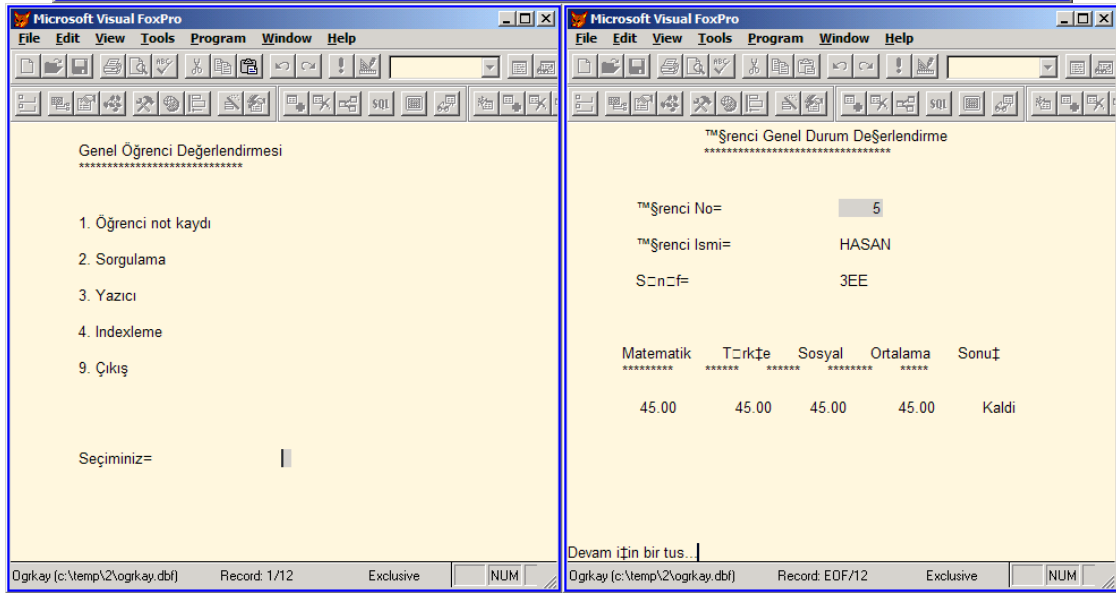
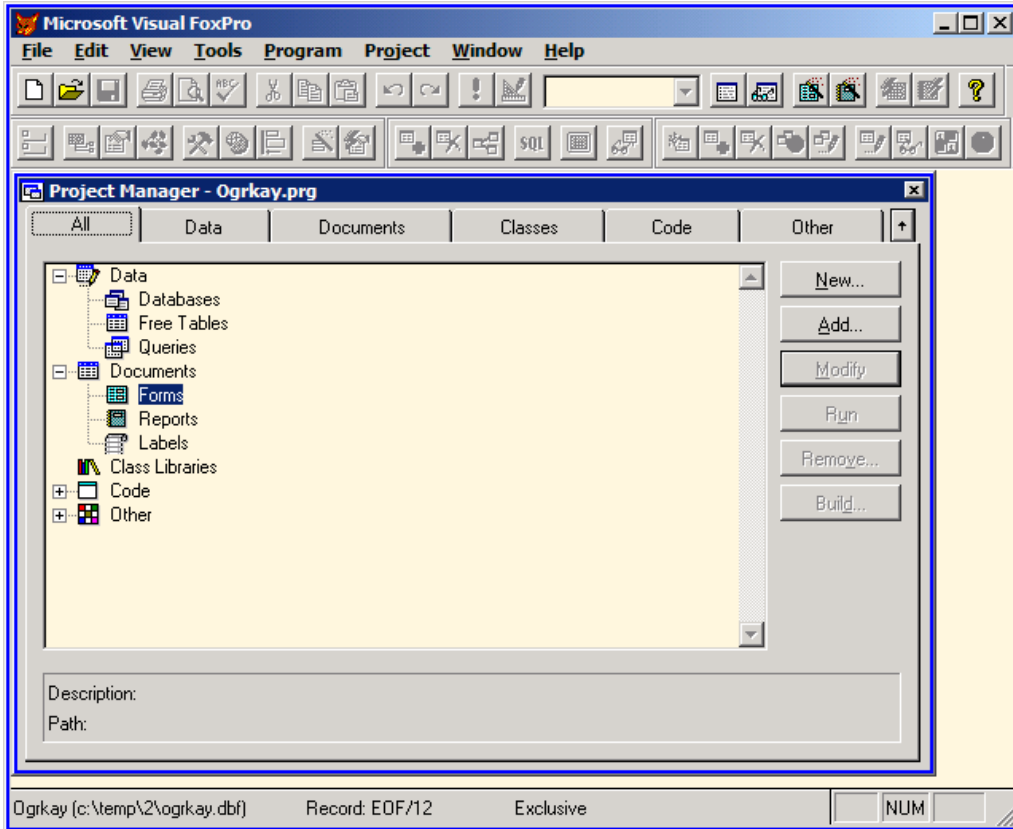
çok if yerine case daha iyi

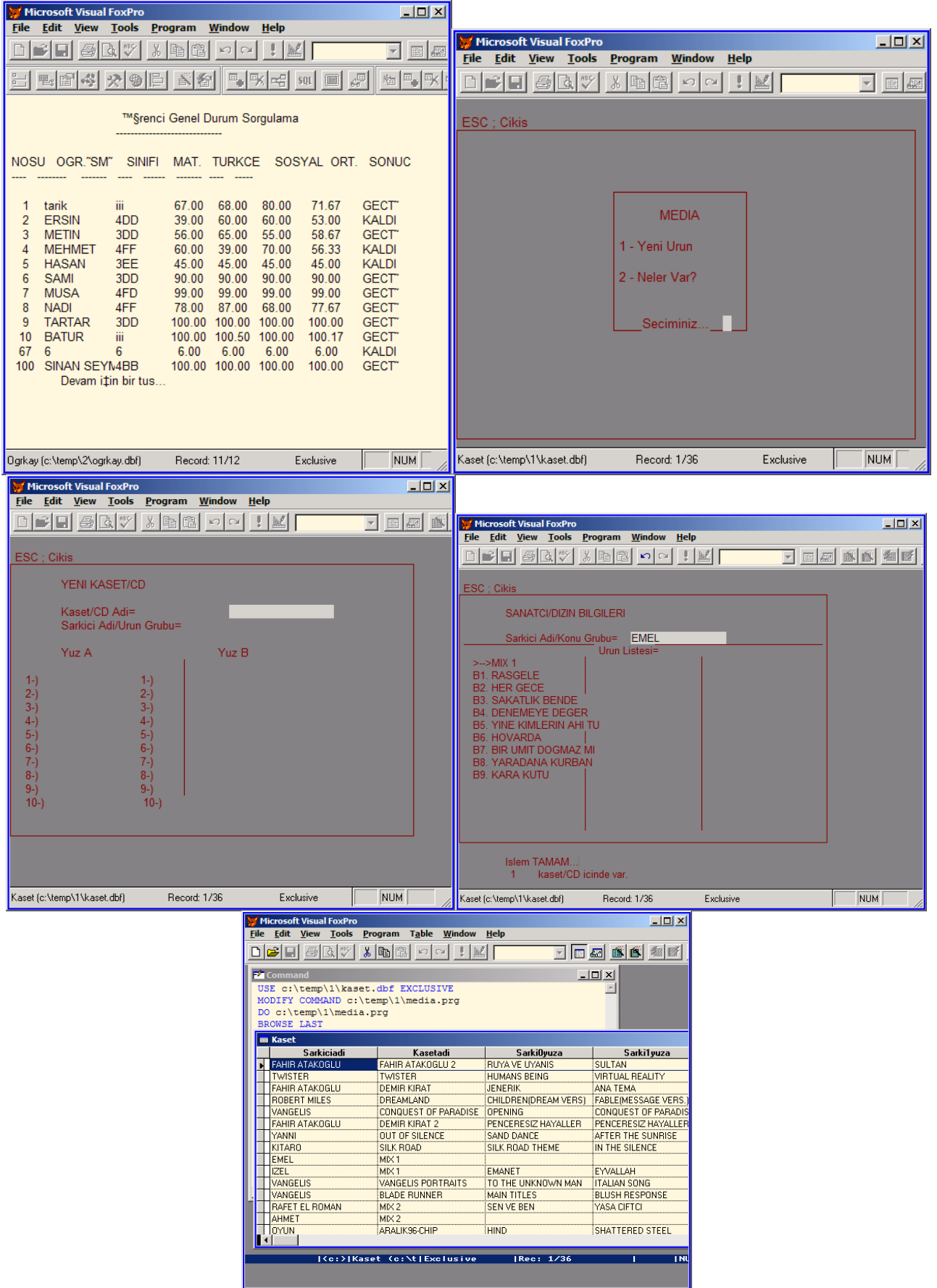
alt program çağırma

döngüye devam

dbase 3 ile Turbo Pascal gibi önce programı hazırlayıp, daha sonra çalıştır komutunu veriyorduk. FoxPro ile de işlemleri menülerden yapıyoruz. Örneğin çalıştırmak için önce *.prg dosya menüsünden seçiliyor, "Program*Do Ctrl+D" ile de program çalıştırılıyor. "View*Browse..." ile de istediğiniz *.dbf veri dosyasını görüntüleme ve düzenleme yapabiliyoruz. Form tasarım sihirbazı ile hoş veri giriş formları ve listeleme, raporlama oluşturabilirsiniz.

Microsoft Access dururken bunlarla neden uğraştık demeyin. dbase Microsoft'tan önce vardı. Şu anda neredeyse tüm programlama dilleri MS Access'i destekliyor. İster internette veritabanı kullanın ister yerel veritabanı olarak kullanın.



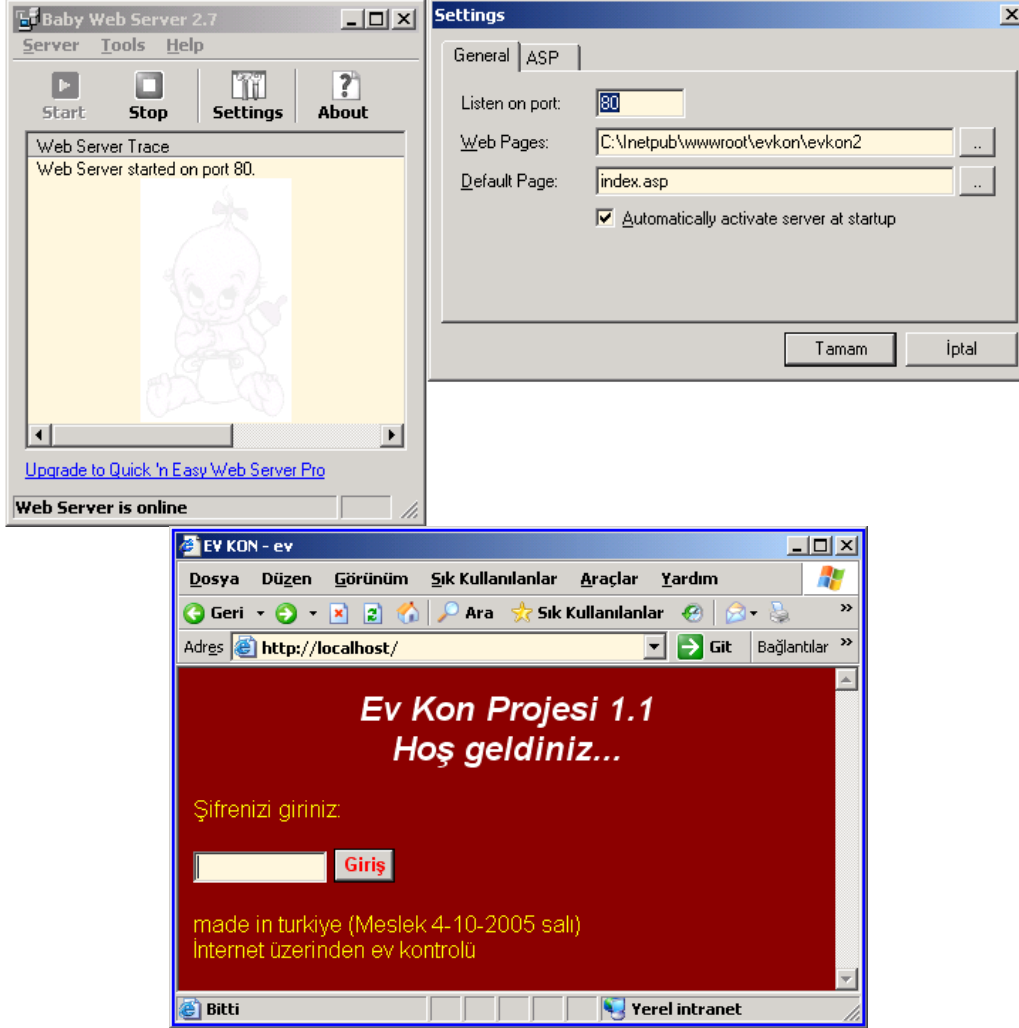


Resim 27 - Görsel dbase

24. Ev Kon (internet üzerinden ev kontrolü)

(Ornekler\Diger\evkon2.zip)

Eski ASP, *.mdb ve VB6 ile çok basit arabirimli bir proje. Tüm bilgisayarlarda (9x veya NT) paralel port kullanarak evdeki aygıtları kontrol ediyoruz. internals.com sitesinden VB ve C için port kullanımı için winio.zip dosyasını indiriniz.

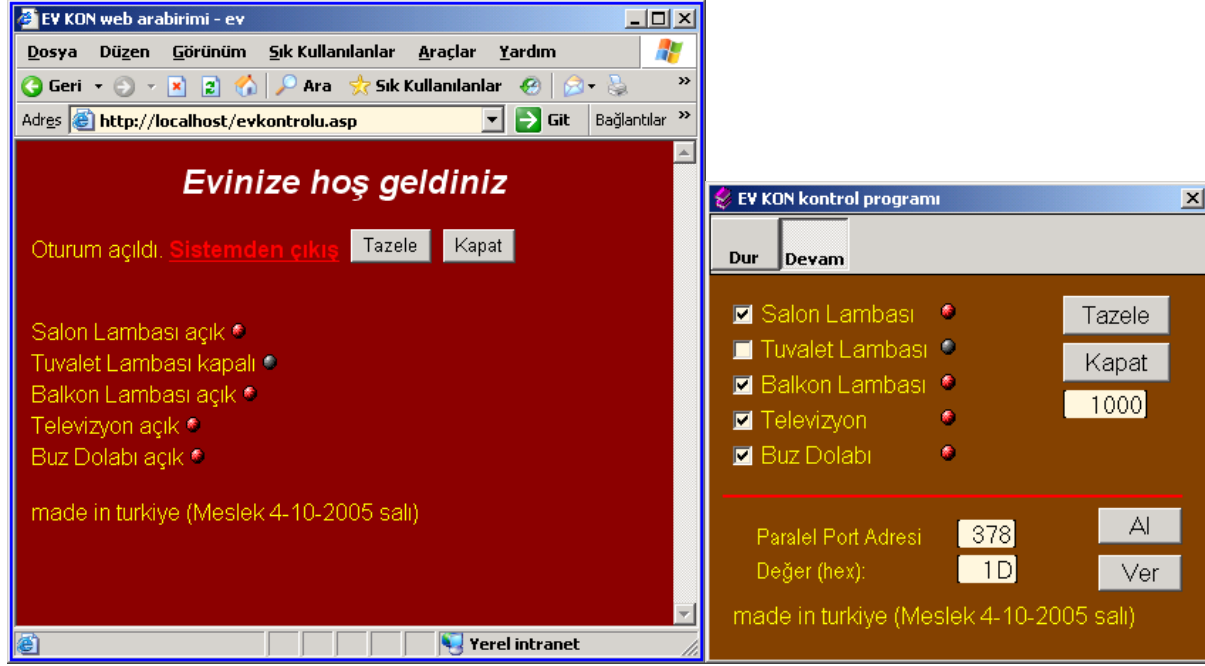


Resim 28a – Ev Kon ekran görüntüsü

Temel fikir; en basit şekilde nasıl ev aletlerini uzaktan kontrol edebiliriz? Web üzerinden donanıma ulaşmak zor olduğundan (Java ve CGI gerekli), aklımıza hemen Delphi ve VB geliyor. ActiveX de kullanılabilirdi. Bu projede ortak veritabanı olarak data.mdb kullanılmasına rağmen, tabii ki MySQL de olabilirdi. Bu projede tamamen Microsoft ürünleri kullanıldı. IIS web yöneticisi yerine bedava ASP destekli Baby Web Server'ı (pablosoftware.com) da tavsiye ederim. Üstteki ilk resimdeki gibi kullanımı çok kolay ve az yer kaplıyor (220 KB).

Sadece 1 günde yaptığım için; güvenlik (şifreli girişteki, veritabanı güvenliği), web kamerası desteği, cep telefonu desteği, Flash ile görsel arabirim gibi birçok eksik var. 5 çeşit ev aleti de çok az. Buna rağmen bilgisayarın portuna takılacak devrenin yetenekleri ile bu ihtimal daha da artırılabilir. Mesela 4 bit seçim adresi, 1 bit de veri olarak düşünülünce $2^4 = 16$ aygıt desteklenebilir.

Alarm, güvenlik sensörleri, belli zamanda belli işlerin yerine getirilmesi, analog veri okuma ve yollama da yapılabilir. Proje harici devre (örneğin bir mikro kontrolör – PIC, PLC gibi) ile PC gücü birleştirilerek çok daha iyi yerlere gelebilir. Sadece paralel port değil, USB ve kablosuz bağlantılar (Bluetooth, kızılötesi - infrared, ses ötesi - ultrasound) da geliştirilebilir. Hırsız alarmı olarak dersek, meydana gelen istenmeyen olaylar SMS ile müşteriye aktarılabilir.



Resim 28b – Ev Kon resimleri, şifre 321.

Bilgisayarınızın ADSL modem ayarlarındaki, 80 portunu dış dünyaya açarak, kendi ana bilgisayar hizmetinizi verebilirsiniz. Yerel ağ için bu ayarı yapmanıza gerek yoktur. Adres kısmına ana bilgisayarın IP'sini girin yeterli.

ASP ve VB artık çok ilkel kalsa da düşük konfigürasyonlu sistemlerde rahat çalışır. Aynı proje PHP, MySQL, Delphi ile de yapılabilir. Ana bilgisayarda port erişimini sağlayan program çalışır halde olmalı, dış dünya da ise sadece gereken o bilgisayarın TCP/IP numarası.

Eğer dinamik IP ile internete bağlanıyorsanız, no-ip.com 'a üye olarak kullanıcı.no-ip.uzanti şeklinde adrese sahip olursunuz. DynIP.com da alternatif. Haftanın 7 günü 24 saat çalışacak makinenizin de ısınma problemi, güç kaynağı problemi ve benzeri problemleri olacaktır. RAM ve sabit diskin kaliteli olması iyi olur.

Kod kısmına bakacak olursak, 2 adet ASP sayfamızda toplam 100 satır kod, tek formulu VB6 projesinde de yaklaşık 100 satırdan oluşuyor. Tüm proje kodları ile sıkıştırılınca 40 KB yapıyor. Hızlı ama kirli de diyebilirsiniz.

Kodlar; VB kodları:

```
Private Sub BtnGet_Click()
Dim Result As Boolean
Dim PortVal As Long
```

```
Result = GetPortVal(Val("&H" + TxtPortAddr), PortVal, 1)
```

```
If (Result = False) Then
MsgBox " There is a problem with GetPortByte.", vbOKOnly + vbCritical, "VBDumpPort32"
Unload FrmVBDumpPort32
Else
```

```
TxtPortByte = Hex$(PortVal And &HFF)
End If
End Sub
```

```
Private Sub BtnQuit_Click()
End Sub
```

```
Private Sub BtnSet_Click()
Dim Result As Boolean
```

```
Result = SetPortVal(Val("&H" + TxtPortAddr), Val("&H" + TxtPortByte), 1)
```



```

If (Result = False) Then
    MsgBox "There is a problem with SetPortByte.", vbOKOnly + vbCritical, "VBDumpPort32"
    Unload FrmVBDumpPort32
End If
End Sub

```

```

Private Sub Check1_Click()
    Data1.Recordset.Edit
    Data1.Recordset.Fields(1).Value = Check1.Value
    Data1.Recordset.Update
    Data1.Refresh
End Sub

```

```

Private Sub Check2_Click()
    Data1.Recordset.Edit
    Data1.Recordset.Fields(2).Value = Check2.Value
    Data1.Recordset.Update
    Data1.Refresh
End Sub

```

```

Private Sub Check3_Click()
    Data1.Recordset.Edit
    Data1.Recordset.Fields(3).Value = Check3.Value
    Data1.Recordset.Update
    Data1.Refresh
End Sub

```

```

Private Sub Check4_Click()
    Data1.Recordset.Edit
    Data1.Recordset.Fields(4).Value = Check4.Value
    Data1.Recordset.Update
    Data1.Refresh
End Sub

```

```

Private Sub Check5_Click()
    Data1.Recordset.Edit
    Data1.Recordset.Fields(5).Value = Check5.Value
    Data1.Recordset.Update
    Data1.Refresh
End Sub

```

```

Private Sub Command1_Click()
    Unload FrmVBDumpPort32
End Sub

```

```

Private Sub Command2_Click()
    Data1.Refresh
    Timer1.Interval = Text1.Text
End Sub

```

```

Private Sub Form_Load()
    If InitializeWinlo = False Then
        MsgBox "There is a problem with InitializeWinlo.", vbOKOnly + vbCritical, "VBDumpPort32"
        Unload FrmVBDumpPort32
    Else
        Call BtnGet_Click
    End If
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Call ShutdownWinlo
End Sub

```

```

Private Sub Timer1_Timer()
    Dim lamba1, lamba2, lamba3, makine1, makine2
    Data1.Refresh

```

```

lamba1 = Data1.Recordset.Fields(1).Value
lamba2 = Data1.Recordset.Fields(2).Value
lamba3 = Data1.Recordset.Fields(3).Value
makine1 = Data1.Recordset.Fields(4).Value
makine2 = Data1.Recordset.Fields(5).Value
If (lamba1 = 1) Then Check1.Value = 1 Else Check1.Value = 0
If (lamba2 = 1) Then Check2.Value = 1 Else Check2.Value = 0
If (lamba3 = 1) Then Check3.Value = 1 Else Check3.Value = 0
If (makine1 = 1) Then Check4.Value = 1 Else Check4.Value = 0
If (makine2 = 1) Then Check5.Value = 1 Else Check5.Value = 0
If (lamba1 = 1) Then Picture1.Visible = 1 Else Picture1.Visible = 0
If (lamba2 = 1) Then Picture2.Visible = 1 Else Picture2.Visible = 0
If (lamba3 = 1) Then Picture3.Visible = 1 Else Picture3.Visible = 0
If (makine1 = 1) Then Picture4.Visible = 1 Else Picture4.Visible = 0
If (makine2 = 1) Then Picture5.Visible = 1 Else Picture5.Visible = 0
If (lamba1 = 1) Then Picture6.Visible = 0 Else Picture6.Visible = 1
If (lamba2 = 1) Then Picture7.Visible = 0 Else Picture7.Visible = 1
If (lamba3 = 1) Then Picture8.Visible = 0 Else Picture8.Visible = 1
If (makine1 = 1) Then Picture9.Visible = 0 Else Picture9.Visible = 1
If (makine2 = 1) Then Picture10.Visible = 0 Else Picture10.Visible = 1

TxtPortByte.Text = Hex(lamba1 * 1 + lamba2 * 2 + lamba3 * 4 + makine1 * 8 + makine2 * 16)
BtnSet_Click
End Sub

```

```

Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)
    If Toolbar1.Buttons(1).Value Then Timer1.Enabled = False
    If Toolbar1.Buttons(2).Value Then Timer1.Enabled = True
End Sub

```

```

Private Sub TxtPortAddr_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        Call BtnGet_Click
    End If
End Sub

```

Kodlar; ASP kodları (evkontrolu.asp):

```

<p align="center" class="style1 style2"><em>
<%
dogruparola = "321"

veritabani = "Driver={Microsoft Access Driver (*.mdb)}; dbq=" & server.mappath("db/data.mdb")
set bag = Server.CreateObject("adodb.connection")
set ks = Server.CreateObject("adodb.recordset")
function temizle(metin)
    metin=replace(metin,chr(10),"") 'satirbasi
    metin=replace(metin,chr(34),"") 'tirnak
    metin=replace(metin,"'", "") 'tirnak
    metin=replace(metin,"<","")
    metin=replace(metin,">","")
    temizle=trim(metin)
end function

if temizle(session("sifre")) = dogruparola then
    girisOK="evet"
end if

If temizle(request("sifre")) = dogruparola or girisOK="evet" then
    session("sifre")=dogruparola
    if temizle(session("sifre")) = dogruparola then
        if request("lamba1")<>"" or request("lamba2")<>"" or request("lamba3")<>"" or request("makine1")<>"" or
request("makine2")<>"" then
            response.Write("Kayıt başladı")
            bag.open veritabani

```

```

sql ="Select * from tablo1"
ks.open sql, bag, 1, 3
if request("lamba1")<>"" then
  ks("lamba1") = request("lamba1")
end if
if request("lamba2")<>"" then
  ks("lamba2") = request("lamba2")
end if
if request("lamba3")<>"" then
  ks("lamba3") = request("lamba3")
end if
if request("makine1")<>"" then
  ks("makine1") = request("makine1")
end if
if request("makine2")<>"" then
  ks("makine2") = request("makine2")
end if
ks.update
ks.close
set ks = nothing
bag.close
set bag = nothing
response.redirect ("evkontrolu.asp")
else
  response.Write("Evinize hoş geldiniz</em>")
  if session("sifre")<>"" then response.write "<p class=style4>Oturum açıldı."
  response.Write(" <a href=index.asp?cikis=evet class=style1 title='Çıkış'>Sistemden çıkış</a>")
  response.Write("&nbsp;&nbsp;&nbsp; <button onclick=location.href('evkontrolu.asp'); value=Tazele
title='Tazele'>Tazele</button>")
  response.Write("&nbsp;&nbsp;&nbsp; <button onclick=window.close(); value=Kapat title='Pencereyi
Kapat'>Kapat</button>")
  bag.open veritabani
  sql = "Select * from tablo1"
  set kayitlar = bag.execute (sql)
  response.write ("<p class=style4>")
  if kayitlar("lamba1") = "0" then
    response.Write ("<br>Salon Lambası kapalı<a href=?lamba1=1><img src=kapali.gif border=0
align=absmiddle></a>")
  else
    response.Write ("<br>Salon Lambası açık<a href=?lamba1=0><img src=acik.gif border=0
align=absmiddle></a>")
  end if
  if kayitlar("lamba2") = "0" then
    response.Write ("<br>Tuvalet Lambası kapalı<a href=?lamba2=1><img src=kapali.gif border=0
align=absmiddle></a>")
  else
    response.Write ("<br>Tuvalet Lambası açık<a href=?lamba2=0><img src=acik.gif border=0
align=absmiddle></a>")
  end if
  if kayitlar("lamba3") = "0" then
    response.Write ("<br>Balkon Lambası kapalı<a href=?lamba3=1><img src=kapali.gif border=0
align=absmiddle></a>")
  else
    response.Write ("<br>Balkon Lambası açık<a href=?lamba3=0><img src=acik.gif border=0
align=absmiddle></a>")
  end if
  if kayitlar("makine1") = "0" then
    response.Write ("<br>Televizyon kapalı<a href=?makine1=1><img src=kapali.gif border=0
align=absmiddle></a>")
  else
    response.Write ("<br>Televizyon açık<a href=?makine1=0><img src=acik.gif border=0
align=absmiddle></a>")
  end if
  if kayitlar("makine2") = "0" then
    response.Write ("<br>Buz Dolabı kapalı<a href=?makine2=1><img src=kapali.gif border=0
align=absmiddle></a>")
  end if

```

```

        else
            response.Write ("<br>Buz Dolabı açık<a href=?makine2=0><img src=acik.gif border=0
align=absmiddle></a>")
        end if
        bag.close
        set bag=nothing
        end if
    end if
else
    response.Write("<p class=style1 >Şifreniz hatalı girildi!")
    session("sifre") = ""
    response.Write("<a href=index.asp?cikis=evet class=style1 title='Çıkış'>Sistemden çıkış</a>")

' response.redirect (Request.ServerVariables("HTTP_REFERER"))
end if
%>
<html>
<style type="text/css">
<!--
.style1 {
    font-family: Arial, Helvetica, sans-serif;
    font-weight: bold;
    color: red;
}
.style2 {
    font-size: large;
    color: white;
}
.style3 {font-family: Arial, Helvetica, sans-serif;
    font-size: middle;
    color: black;
}
.style4 {font-family: Arial, Helvetica, sans-serif;
    color: yellow;
}
-->
</style>
</head>

<body bgcolor=darkred>
<title>EV KON web arabirimi</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
</head>

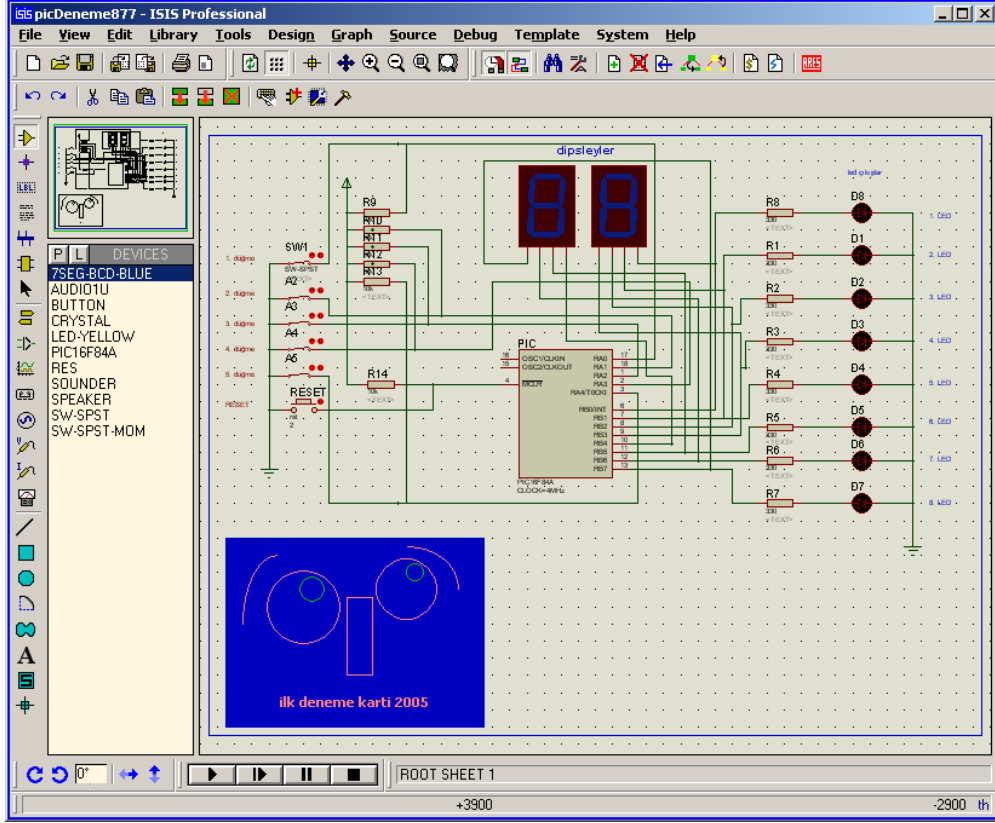
<p class="style4">made in turkiye (Meslek 4-10-2005 salı)
</p>
</body>
</html>

```

25. PIC (Peripheral Interface Controller)

(Ornekler\Diger\evkon2.zip)

Nihayet doğrudan donanım (RISC işlemci Microchip PIC) üzerine projemiz de var. Şimdilik genel olarak ne olduğunu gösterelim. microchip.com ana sitesinden gerekli doküman ve program olan MPLAB indiriniz. Sadece assembly ile de PIC programlanabilir, veya PICBasic, PIC-C, JAL gibi yüksek seviyeli diller ile de program yazabiliriz. Denemek için de Labcenter ISIS Proteus ile "deneme kartı" yapıp, simüle edebiliriz. Hatâları giderdikten sonra, son adım olarak da oluşan *.hex dosyayı gerçek PIC'e aktarmak (programlamak) kalıyor.



Resim 29a – Proteus ile ilk deneme kartımız 16f84

İnternet üzerinde kısa bir araştırmadan sonra, PIC ile ilgili birçok doküman ve program bulabiliriz. Mikro kontrolör olarak PIC'in seçilmesinin nedeni ucuz ve genişlemeye açık yapıda olmasıdır. Daha ileri seviyede yani 16f84(18 pic) yetmediği durumlarda, 16f628(18 pin) veya 16f877(40 pin) kullanılabilir. Aynı komutları (35 adet) bu işlemcide de rahatlıkla kullanabiliriz. Hem hızlı bir işlemcidir (20Mhz'de 5 milyon işlem), hem de piyasada kolay bulunur.

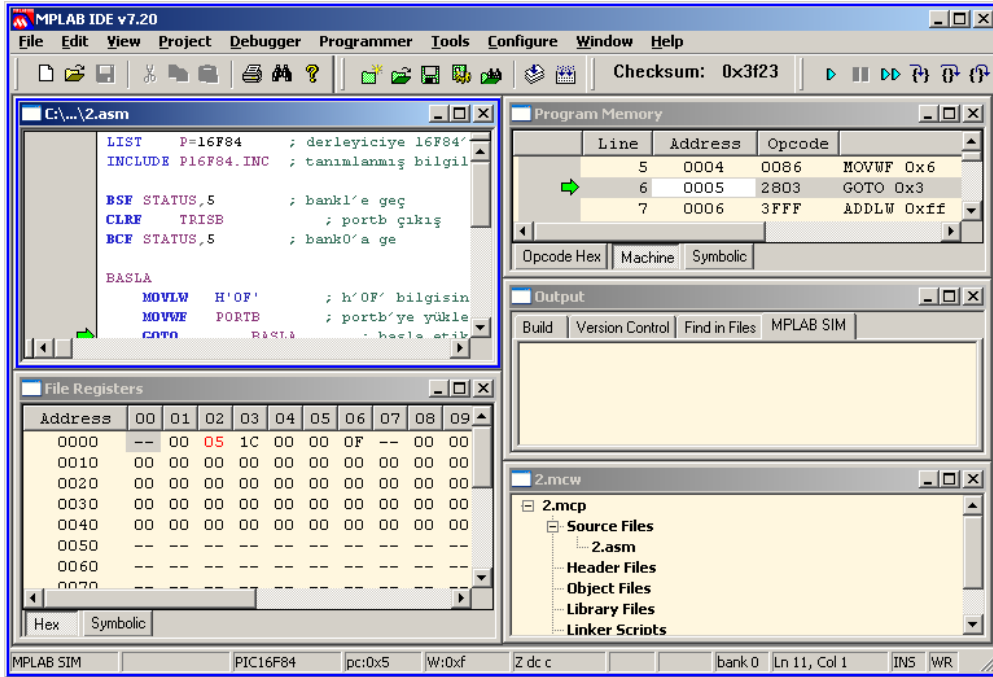
Programcılar uzun süre sadece assembler kodları ile programlama yapmışlar, son yıllarda da aynen PC'deki gelişme yolu gibi, yüksek seviyeli konuşma diline benzer diller ile de PIC kodlama mümkün olmuştur. PICBIT ile PLC'deki görsel tasarım gibi PIC'de de program yazabiliriz. Ayrıca yapılan programın çalışıp çalışmadığını bilgisayarda da test edebilme imkânımız vardır. uSim ve uEditor ile 877 haricindeki PIC'leri programlayıp deneyebiliriz.

PIC'e aktarım olarak PICup gibi birçok bedava program bulabilirsiniz. PIC Start Plus ise hazır bir EPROM programlama donanımdır. COM yani seri port dışına LPT (yazıcı) ve USB ile de iletişim kurulabilir, programlanabilir.

Neredeyse birçok program, düzenleyici, derleyici, çevirici, örnek, doküman, devre şeması bedava olarak bulabiliyoruz. Sadece Windows değil, Linux ve Macintosh ile de PIC programlama yapılabilir.

Genelde bir fabrikanın otomasyonu üzerine çalışmalar yapılabilirken, günlük hayatı kolaylaştırıcı robotlar, güvenlik sistemleri, müzikli alarm, saat gibi birçok proje geliştirilebilir. PC'den bağımsız çalışan, sadece açma düğmesine basılarak çalışan basit arabirimden tutun, gelişmiş LCD, radyo frekanslı arabirimli, sensörlerle donatılmış karmaşık devreler de yapılabilir.

"Ornekler\Diger\merdiven_oto.zip" içinde Proteus ve PicBasicPro ile yazılmış küçük bir projemiz var. En altta bununla ilgili kısma ve resimlere ulaşabilirsiniz.



Örnek *.asm programımız (okulumuz öğretmenlerinden İhsan KAR'ın örneği):

;A1'e basıldığında sayı kaydedicisine 3 yükleyen
 ;a2'ye basıldığında 7 yükleyen
 ;a3'e basıldığında 255 yükleyen
 ;a0'a basıldığında yüklenen değerleri portb'de gösteren program

```

LIST P=16F84
INCLUDE P16F84.INC
SAYI EQU H'0C'
BSF STATUS,5 ;bank1'e geç
CLRF TRISB ;portb çıkış
MOVLW H'FF'
MOVWF TRISA ;porta giris
BCF STATUS,5 ;bank0 a gec
CLRF PORTB ;portb'yi temizle

BASLA
TEST1
    BTFSC PORTA,0
    GOTO TEST2
    CALL GOSTER
    GOTO BASLA

TEST2
    BTFSC PORTA,1
    GOTO TEST3
    CALL YUKLE3
    GOTO BASLA

TEST3
    BTFSC PORTA,2
    GOTO TEST4
    CALL YUKLE7
    GOTO BASLA

TEST4
    BTFSC PORTA,3
    GOTO BASLA
    CALL YUKLE255
    GOTO BASLA
  
```

GOSTER

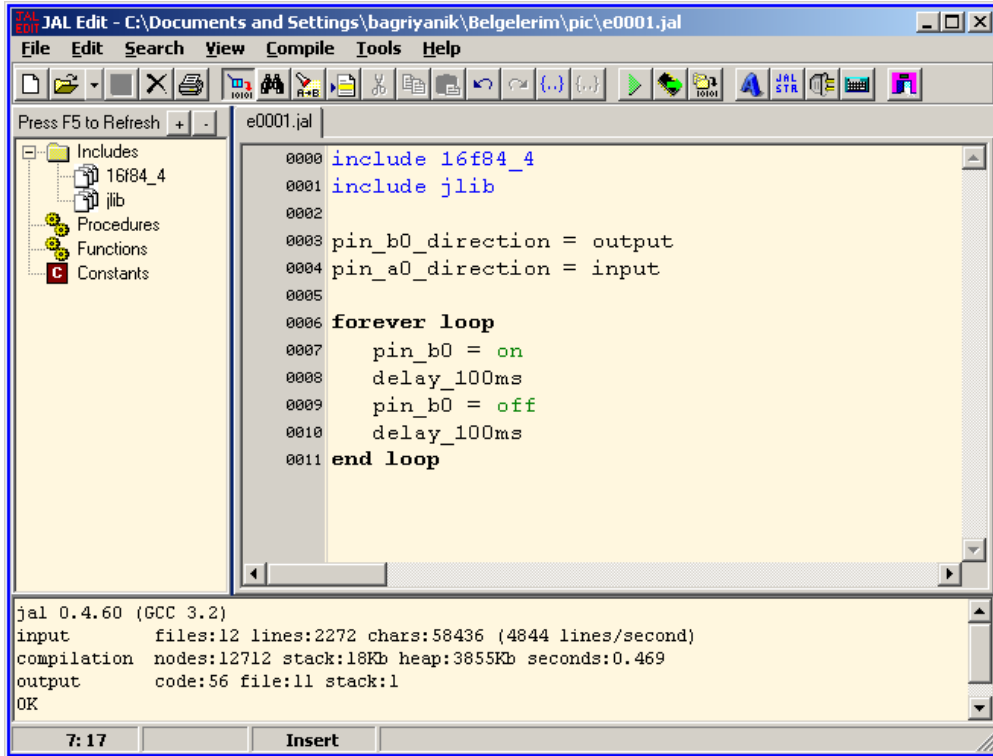
```
MOVWF SAYI,W
MOVWF PORTB
RETURNYUKLE3
MOVLW H'03'
MOVWF SAYI
RETURN
```

YUKLE7

```
MOVLW H'07'
MOVWF SAYI
RETURN
```

YUKLE255

```
MOVLW H'FF'
MOVWF SAYI
RETURN
END
```



Örnek bir *.jal JAL(Just another language, örnek JAL Edit programından alındı, JAL tamamen bedava (GNU) bir dildir, pascal'a benzer) programı:

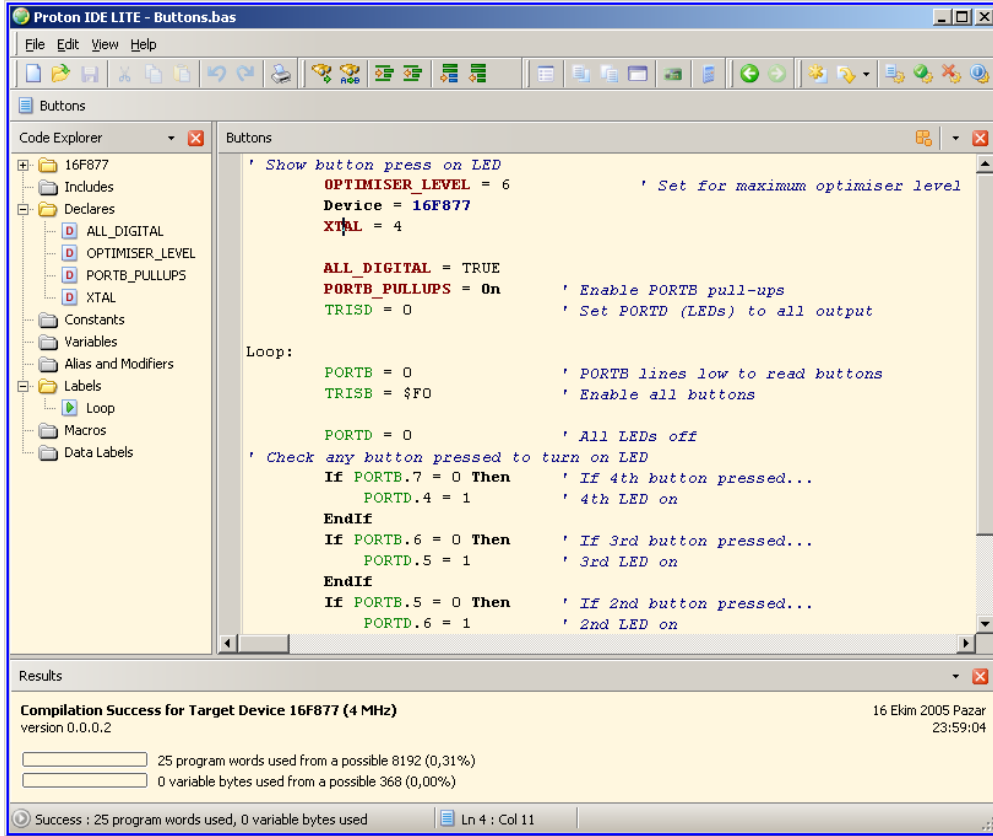
```
include 16f84_4
include jlib
```

```
pin_b0_direction = output    --b0 pini çıkış
pin_a0_direction = input    --a0 pini giriş
```

```
forever loop                --sonsuz döngü
  pin_b0 = on                --b0 çıkışı ON
  delay_100ms                --100 milisaniye gecikme
  pin_b0 = off
  delay_100ms
end loop
```

Birçok programlama dili alternatifimiz ve programlama editörümüz var. En basit dil Basic ve en basit editör de not defteri. En zor dil assembler ve en karışık arabirim de MLABIDE. Ortak olan konu ise ISIS Proteus. Yaptığımız projeyi PIC'e yüklemeyen önce bilgisayarda deneyebiliyoruz. Hem dijital, hem de elektronik deneylerin tümünü bu program ile deneyebiliriz. Electronics Workbench ise eski bir program artık. Sırf PIC için yapılmış simülatörler olsa da, Proteus'un yanına yaklaşıyorlar. Osiloskop, ölçü aleti, sinyal jeneratörü gibi çok hoş eklentileri var.

Başka bir programlama yöntemi de sadece Proteus'un derlemesine bırakarak *.asm kodlamadır. Zaten Proteus içinde MPASM derleyicisi vardır. Ama 35 adet PIC komutunu iyice öğrendikten sonra istediğiniz programı yapabilir hale geliyorsunuz. Asm her ne kadar karmaşık gözükse de, küçük *.hex dosyası olduğu için ve programcıya tam kontrol imkânı vermesi sebebi ile tercih edilir. Zaman kısıtlamanız yok ise *.asm ile uğraşınız. Ama kısa sürede proje geliştirmeniz gerekirse, PBP (PicBasicPro) en iyisi. Mesela merdiven otomatığını hiç *.asm'den anlamadan 1 saat gibi kısa sürede yaptım, ve Proteus'da çalıştırdım. 16f877 kullandım. PortA, B, C, D ve E gibi birçok giriş çıkış özelliği var. 16f84'de de Port A ve B var. Biraz uğraşarak 84 ile de istediğimizi yapabiliriz. 16f84 artık üretilmiyor, bu arada.



Örnek bir *.bas PicBasic (ProtonIDELite örneklerinden alındı – demo olduğundan 50 satır derleyebilir):

```
' Show button press on LED
OPTIMISER_LEVEL = 6
Device = 16F877
XTAL = 4

ALL_DIGITAL = TRUE
PORTB_PULLUPS = ON      ' Enable PORTB pull-ups
TRISD = 0                ' Set PORTD (LEDs) to all output

Loop:
PORTB = 0                ' PORTB lines low to read buttons
TRISB = $F0              ' Enable all buttons

PORTD = 0                ' All LEDs off
' Check any button pressed to turn on LED
If PORTB.7 = 0 Then      ' If 4th button pressed...
    PORTD.4 = 1          ' 4th LED on
EndIf
If PORTB.6 = 0 Then      ' If 3rd button pressed...
    PORTD.5 = 1          ' 3rd LED on
EndIf
If PORTB.5 = 0 Then      ' If 2nd button pressed...
    PORTD.6 = 1          ' 2nd LED on
```



```

Endif
If PORTB.4 = 0 Then ' If 1st button pressed...
    PORTD.7 = 1    ' 1st LED on
Endif
Goto Loop          ' Do it forever

```

The screenshot shows the PicBasic Pro IDE interface. The main window displays the following code:

```

trisa=%00011111      ' port a giriş
trishb=%00000000
trisc=%00000000
trisd=%00000000
trise=%00000000

adcon1= 7             'digital input for A

porta=0:portb=0:portc=0:portd=0:porte=0

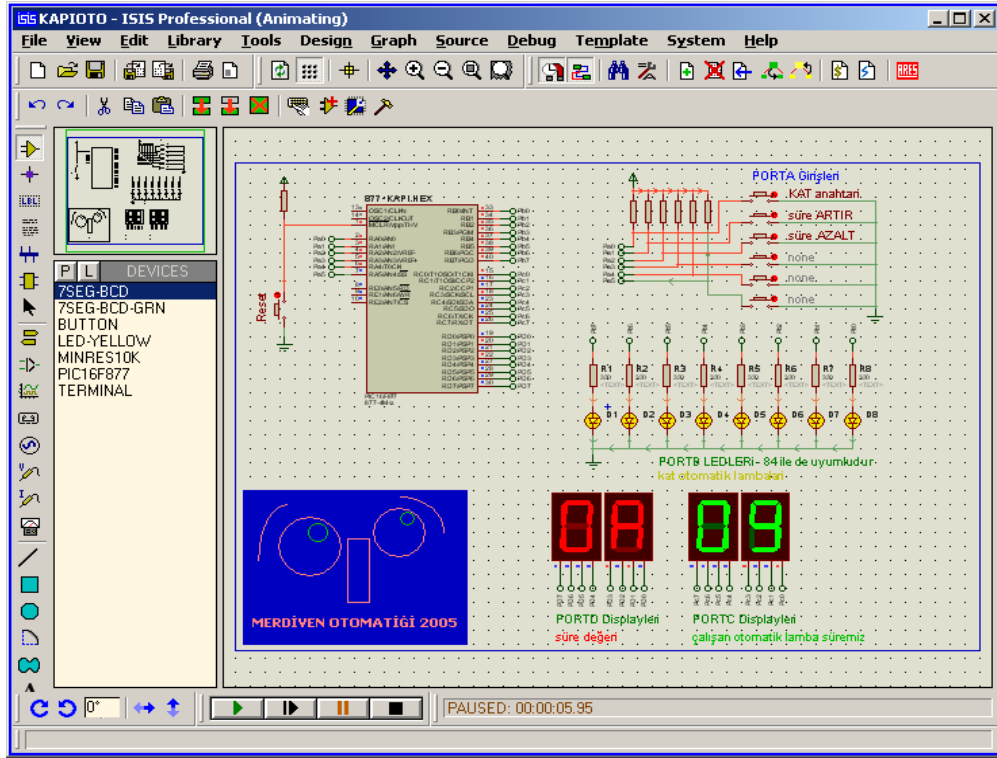
sure      VAR    BYTE
sure = 10          'varsayılan süre = 30 sn
asilsure  VAR    BYTE
asilsure = 10      'varsayılan süre = 30 sn
islem     VAR    BIT
islem = 0          'işlem hemen başlamasın
hak       VAR    BIT
hak = 0           'süre uzatma hakkı 1

portb.0 = 0         'kat lambaları

```

The status bar at the bottom indicates: Line: 1 Col: 1. The bottom panel shows compilation statistics: 340 words used, PicBasic Pro Compiler 2.43, (c) 1998, 2002 microEngineering Labs, Inc. All Rights Reserved. C:\jalcc\PBPro>pm.exe kapi PM Assembler 4.05, Copyright (c) 1995, 2002 microEngineering Labs, Inc. 340 words used.

Resim29b - PicBasicPro dilinin Pic Mate programı ile derlenmesi.



Resim29c - Merdiven otomatığı, süre kontrollü ve ek süre de istenebiliyor

Merdiven otomatığını programlama ve test etmesi sadece 1 saatimi aldı. Eğer alt seviye olan *.asm ile yapsaydım herhalde 1 hafta sürerdi. Tabi *.asm kodlarını iyice öğrendikten sonra bu süre daha da kısalabilir. PicBasic kodları ise aşağıdadır. Yaklaşık 60 satır. Programın eksiği 7 parçalı led göstergelerde onaltılık sayı sisteminde sürelerin gösterilmesi. Yani günlük hayatta bu proje kullanılacaksa, göstergede onluk sayı olarak süre gösterilmelidir. Süre değeri 0A demek aslında 10 saniye demek.

Port B'nin tüm 8 biti de kat lambalarına bağlı. 8 kata kadar problem yok, ama aslında sadece 1 çıkış da bizim işimizi görürdü. Kat anahtarının sadece 1 adet olması gibi. Kat anahtarı ve lambaları istediğimiz kadar, dış devrede çoğaltabiliriz.

Küçük bir proje olsa bile neredeyse tüm portları kullandık. Daha çok giriş ve çıkış gerekseydi, biz de başka yollara başvurabiliriz. Mesela bir port belli bir zaman giriş, belli bir zaman da çıkış olabilir, ya da tüm display'ler tek bir yerden sıra ile sürülebilir. Bu tür numaralarda hem yazılım olarak hem de donanım olarak çözümler üretebiliriz. Eğer çare kalmamışsa, birden fazla PIC kullanılabilir. Tam tersine 1 PIC çok geliyor ise, o zaman da içine birden fazla program yükleyerek, tek merkezden hızlı bir şekilde program güncelleme, başka yardımcı işler üstünlüğü kazanabiliriz. Mesela bir saat programında, alarm, süreölçer gibi birçok program bir yerden çalışabilir. Alarmede çalacak müzik de başka bir yerde saklanabilir. Teknik konulara fazla girmiyorum biliyorsunuz.

```
trisa=%00011111
trisb=%00000000
trisc=%00000000
trisd=%00000000
trise=%00000000
```

' port a giriş, geri kalan portlar çıkış

```
adcon1= 7
porta=0:portb=0:portc=0:portd=0:porte=0
sure VAR BYTE
sure = 10
asilsure VAR BYTE
asilsure = 10
islem VAR BIT
islem = 0
hak VAR BIT
hak = 0
```

'digital input for A, normalde analog giriştir

'varsayılan süre = 30 sn

'varsayılan süre = 30 sn

'işlem hemen başlamasın

'süre uzatma hakkı 1

```

        portb.0 = 0          'kat lambaları
        portb.1 = 0
        portb.2 = 0
        portb.3 = 0
basla:
    if porta.1 = 0 and asilsure<$ff and islem = 0 then 'sure 1 sn artar
        asilsure= asilsure + 1
        pause 200          'yeter BOUNCE
    endif
    if porta.2 = 0 and asilsure>0 and islem = 0 then 'sure 1 sn azalır
        asilsure= asilsure - 1
        pause 200          'yeter BOUNCE
    endif

    portc = sure
    portd = asilsure

    if (porta.0 = 0) and (islem = 0)and(asilsure<>0) then
        islem = 1
        sure = asilsure
        pause 200          'yeter BOUNCE
    endif

    if (porta.0 = 0) and (islem = 1) then          'ek sure de yapabiliyoruz
        if hak = 0 then hak = hak + 1
    endif

    if islem = 1 then          'işlem başlamışsa
        portb = %11111111    'kat lambaları
        Pause 1000
        sure = sure - 1
    endif

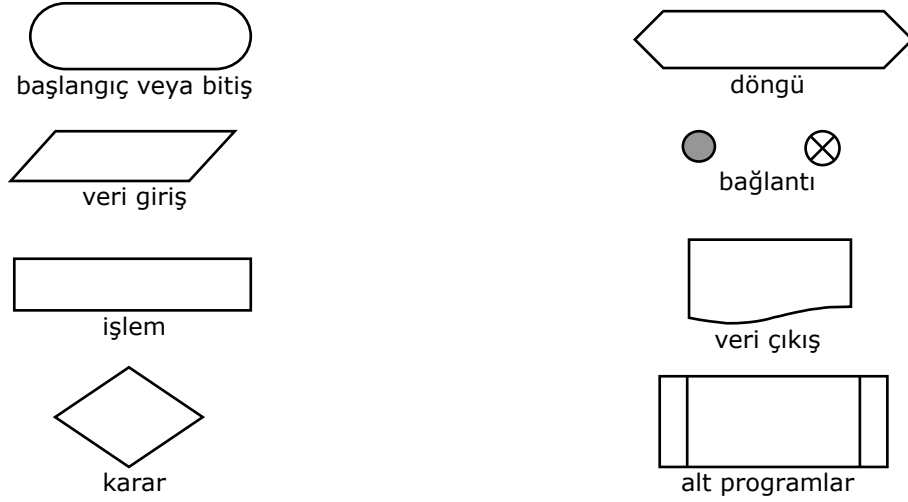
    if sure = 0 then          'süre bitmişse
        if hak = 1 then      'süre uzatma hakkım var
            sure = asilsure
            hak = 0
        else
            portb = %00000000    'kat lambaları
            islem = 0
            sure = asilsure
            hak = 0
        endif
    endif

    goto basla
end

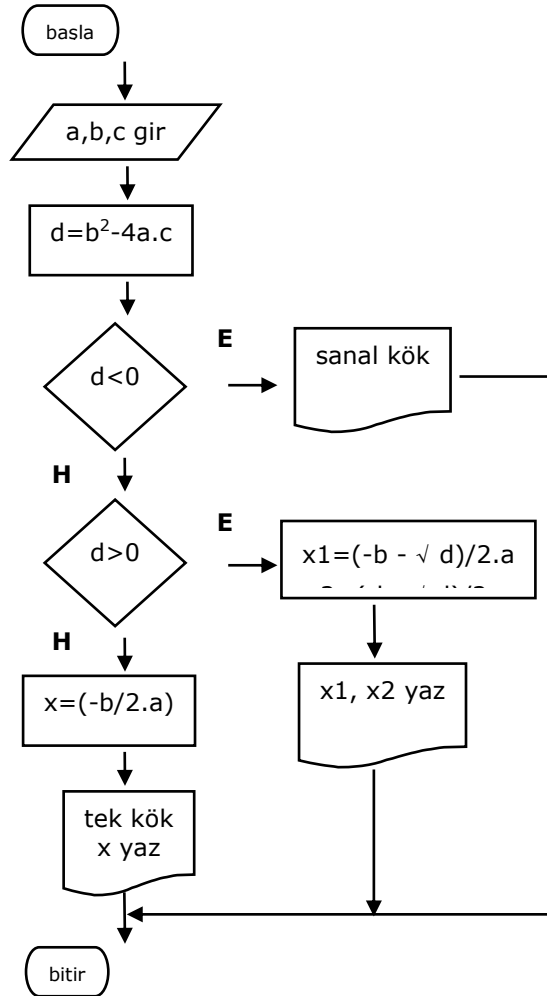
```

Ek 1 – Algoritma Oluşturmak

Bir problemin çözümünde yardımcı, ana hatları belirten doküman veya açıklayıcı nota algoritma denir. Programlama dillerindeki gösterimi akış diyagramı veya çizelgesidir. Her dile uygulanabilen genel uluslararası sembollerle çizimi yapılır. Bir dilden diğerine algoritma yaparak geçebilirsiniz.



İkinci derece denklem ($ax^2+bx+c=0$) köklerini (x1, x2 veya x) bulmak için şu algoritma kullanılabilir:



Ek 2 – SQL

Delphi veritabanı nesnelerinden Query – sorgu içinde kullandığımız SQL özelliği hakkında basit birkaç ipucu veriyorum. Aynı şekilde Microsoft Access ve dbase içinde de SQL - (Structured Query Language – Yapısal Sorgulama Dili) kullanılıyor. Oracle, InterBase, MSSQL Server ve MySQL de bu dil için yazılmış ağ yazılımlarıdır.

SQL ile bir tablo filtrelenebilir, arama yapılabilir, belli kayıtlar silinebilir veya güncellenebilir, tablolar ilişkilendirilebilir, hesaplanan alanlar oluşturulabilir, kayıtlar sıralanabilir.

Örnekleri "Ornekle\Delphi\SQL" dizinindeki program ile deneyebilirsiniz. İçinde tek tırnak veya çift tırnak ile değer belirtebilirsiniz. Programda sadece üstteki ilk açılır kutudan DBDEMOS seçip aşağıdaki örneklerden istediğinizi *select * from tablo* yazan açılır kutu içine yazmanız yeterlidir.

En çok kullandığımız deyimler select, from ve where'dir.

select : * ile tüm alanları veya belli alan adları seçeriz.
from : tablo ad(lar)ı belirtilir
where : kriter belirtilir, where kullanılmazsa tüm kayıtlar görüntülenir.

Örnekler:

select * from country
 country.db'deki tüm kayıtları getirir. Böylece Query ile Table aynı etkiye sahip olur

select Name, Capital from country where population > 2000000

2.000.000'dan büyük nüfuslu başkentlerin ad ve isimlerini alır

select * from country where continent = "South America" and population > 2000000

Güney Amerikadaki 2.000.000 üstü nüfuslu ülkelerin listesi

select distinct Continent from country

Sadece kıta adlarını gösterir

select orders.orderno, orders.saledate, customer.company, customer.contact, employee.lastname from orders, customer, employee where orders.custno = customer.custno and orders.empno = employee.empno

Siparişi alan çalışan ve siparişi veren müşteri üç tablo birleştirilerek, ekrana beş alanı listeliyor

select * from country order by continent, name

Kıta adına sonra da ülke adına göre tablo sıralanıyor

select sum (orders.amountpaid), count (orders.amountpaid), avg (orders.amountpaid) from orders, employee where orders.empno = employee.empno and employee.lastname = "Johnson"

Johnson soyadlı elemanın toplam, sayı ve ortalama aldığı sipariş miktarları listelenir

select employee.lastname, sum (orders.amountpaid) from orders, employee where orders.empno = employee.empno group by employee.lastname

Her elemanın aldığı toplam sipariş miktarını verir

select firstname,lastname from employee where firstname like "%nn%" and lastname like "%en%"

Adında "nn" ve soyadında "en" geçen personeli listeler, tabii ki her kelimenin baş ve sonunda % kullanmak zorunda değilsiniz, "An%" gibi de olabilir (case sensitive – büyük küçük harf duyarlı)

Ek 3 – ASCII Kod Tablosu

(American Standard Code for Information Interchange)

Askii diye okunur. 256 yazı ve kontrol karakterinden oluşan bu tablonun ilk 128'si sabit, kalan 127 karakter yazıcı, bilgisayar ve dile göre değişir.

Dec	Hex	Ctrl	Adı	Anlamı	Dec	Hex	Ctrl	Adı	Anlamı
0	00	^@	NUL	null (end string)	16	10	^P	DLE	data line escape
1	01	^A	SOH	start of heading	17	11	^Q	DC1	dev ctrl 1 (X-ON)
2	02	^B	STX	start of text	18	12	^R	DC2	device ctrl 2
3	03	^C	ETX	end of text	19	13	^S	DC3	dev ctrl 3 (X-OFF)
4	04	^D	EOT	end of transmission	20	14	^T	DC4	device ctrl 4
5	05	^E	ENQ	enquiry	21	15	^U	NAK	negative acknowledge
6	06	^F	ACK	acknowledge	22	16	^V	SYN	synchronous idle
7	07	^G	BEL	bell	23	17	^W	ETB	end transmit block
8	08	^H	BS	backspace	24	18	^X	CAN	cancel
9	09	^I	HT	TAB horizontal tab	25	19	^Y	EM	end of medium
10	0a	^J	LF	line feed	26	1a	^Z	SUB	substitute
11	0b	^K	VT	vertical tab	27	1b	^[	ESC	escape
12	0c	^L	FF	form feed	28	1c	^\	FS	file separator
13	0d	^M	CR	carriage return	29	1d	^]	GS	group separator
14	0e	^N	SO	shift out	30	1e	^^	RS	record separator
15	0f	^O	SI	shift in	31	1f	^_	US	unit separator

ASCII Kod Tablosu:

0	32	[space]	64	@	96	`
1	33	!	65	A	97	a
2	34	"	66	B	98	b
3	35	#	67	C	99	c
4	36	\$	68	D	100	d
5	37	%	69	E	101	e
6	38	&	70	F	102	f
7	39	'	71	G	103	g
8	* *	(	72	H	104	h
9	* *	)	73	I	105	i
10	* *	*	74	J	106	j
11		+	75	K	107	k
12		,	76	L	108	l
13	* *	-	77	M	109	m
14		.	78	N	110	n
15		/	79	O	111	o
16		0	80	P	112	p
17		1	81	Q	113	q
18		2	82	R	114	r
19		3	83	S	115	s
20		4	84	T	116	t
21		5	85	U	117	u
22		6	86	V	118	v
23		7	87	W	119	w
24		8	88	X	120	x
25		9	89	Y	121	y
26		:	90	Z	122	z
27		;	91	[	123	{
28		<	92	\	124	
29		=	93	]	125	}
30		>	94	^	126	~
31		?	95	_	127	

128	160	[space]	192	À	224	à
129	161	ı	193	Á	225	á
130	162	¢	194	Â	226	â
131	163	£	195	Ã	227	ã
132	164	¤	196	Ä	228	ä
133	165	¥	197	Å	229	å
134	166	¦	198	Æ	230	æ
135	167	§	199	Ç	231	ç
136	168	¨	200	È	232	è
137	169	©	201	É	233	é
138	170	ª	202	Ê	234	ê
139	171	«	203	Ë	235	ë
140	172	¬	204	Ì	236	ì
141	173	-	205	Í	237	í
142	174	®	206	Î	238	î
143	175	¯	207	Ï	239	ï
144	176	°	208	Ğ	240	ğ
145	177	±	209	Ñ	241	ñ
146	178	²	210	Ò	242	ò
147	179	³	211	Ó	243	ó
148	180	´	212	Ô	244	ô
149	181	µ	213	Õ	245	õ
150	182	¶	214	Ö	246	ö
151	183	·	215	×	247	÷
152	184	¸	216	Ø	248	ø
153	185	¹	217	Ù	249	ù
154	186	º	218	Ú	250	ú
155	187	»	219	Û	251	û
156	188	¼	220	Ü	252	ü
157	189	½	221	Ý	253	ý
158	190	¾	222	Ş	254	ş
159	191	¿	223	ß	255	ß

Alt tuşuna basarken mesela 65 rakamına basıp Alt tuşunu bırakırsanız ekrana "A" yazılır. MS-DOS ve Windows'un yazı karakterleri farklıdır. 200 ncü karakter DOS'ta "L" iken Windows'ta "+" olarak çıkabilir. Enter, Return, Cr – Carriage Return aynı anlamda kullanılıyor. Fakat nümerik klavyedeki ve Backspace tuşu altındaki Enter ayrı tuş kodu ile çalışır. Bunun sebebi nümerik klavyedeki Enter'ın ayrıca 0xe0 değeri üretmesidir. Bazı tuşlar (Ctrl, Shift, Alt, Fn,...) tek kod üretmezler. "CD2\Programlama\HelpPC" dizinindeki programlardan ayrıntıları elde edebilirsiniz. Bizim için en önemli karakterlerden bazıları 13 Enter, 27 ESC tuşudur. VB dilinde vbCr (vbEnter değil), vbTab, vbCrLf,... ; Delphi'de VK_TAB, VK_RETURN, VK_ESCAPE gibi hazır sabitler vardır.

MsgBox ("Adı:" & vbTab & "Ahmet")

ShowMessage('Adı:' + chr(VK_TAB) + 'Ahmet');

'VB'de tuş kullanımı

//Delphi'de tuş kullanımı

Bu arada sık kullandığımız bazı özel karakterlerin İngilizce yazım, okunuş ve Türkçe anlamları da şunlardır:

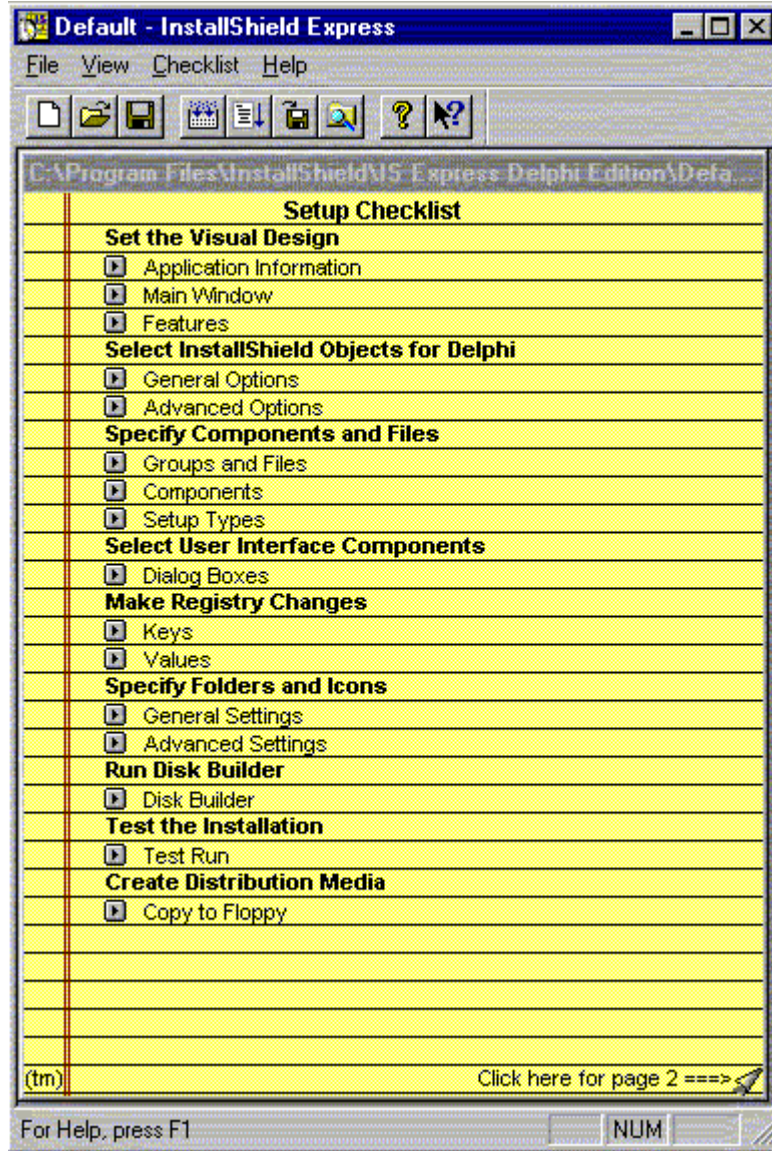
boşluk	space (<i>speys</i>)	?	soru	question mark (<i>kuesçın mark</i>)
"	çift tırnak	*	yıldız	asterisk (<i>estırıks</i>)
'	tek tırnak	,	virgül	comma (<i>kama</i>)
!	ünlem	;	noktalı virgül	semicolon (<i>semikalın</i>)
^	şapka	:	iki nokta üst üste	colon (<i>kalin</i>)
#	diyez	.	nokta	dot, period (<i>pirid</i>)
+	artı	@	güzel a	at sign (<i>et sayn</i>)
-	tire, eksi	~	tırtıl	tilde (<i>tayld</i>)
\$	dolar	/	bölme	division (<i>divijın</i>), slash (<i>sleş</i>)
%	yüzde	\	ters bölme	back slash (<i>bek sleş</i>)
&	ve		çubuk, boru	pipe (<i>payp</i>)
{ }	küme	_	alt çizgi	underline (<i>andı:layn</i>)
()	parantez	-	çizgi	en dash (<i>en deş</i>)
[]	köşeli parantez	-	çizgi	em dash (<i>em deş</i>)
=	eşittir	<	küçüktür	less than (<i>les den</i>)

Ek 4 – InstallShield’in Kullanımı

InstallShield Express for Delphi 3.0 Programı Kullanımı

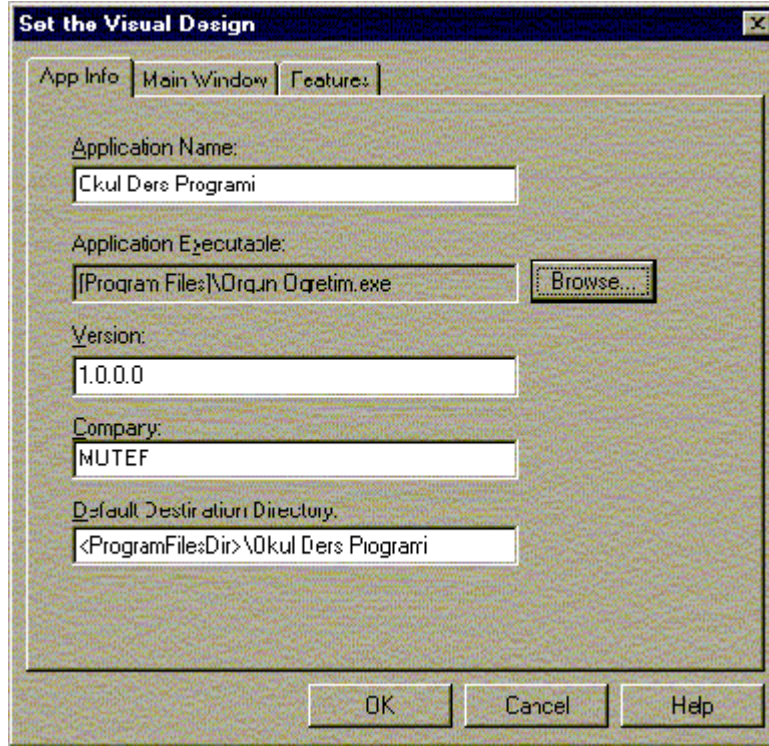
Okul Ders Programının kurulum disketleri haline getirilmesi InstallShield Express for Delphi 3.0 programı ile gerçekleştirilmiştir. İstedığınız programın kurulumunu bu adımlarla hazırlayabilirsiniz.

Yeni sürüm InstallShield Express’in arabirimi daha iyileşirken genel proje yapısı benzer. *.iwz uzantılı eski projelerinizi Delphi 6 veya üstü paketlerin yanında gelen kurulum programı ile de açabilirsiniz. Yeniliklerin başında olan web entegrasyonunun daha fazla olduğu dikkatinizi çekebilir. Sadece Delphi’de değil VB gibi dillerle yaptığınız programın Access, BDE, ODBC, INI, dosya ilişkilendirmeleri, registry gibi ince ayarlarını rahatlıkla yapabilirsiniz.



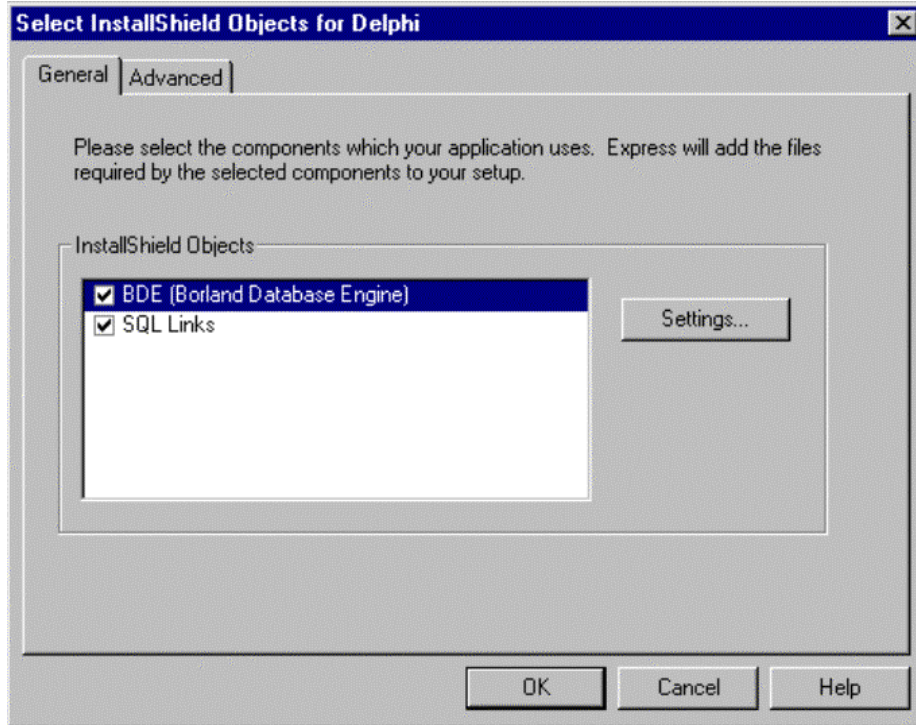
Resim 1 – InstallShield Express 1.11a versiyonunun görüntüsü

Delphi 3.0’ın yanında gelen bu program yardımı ile proje veya herhangi bir uygulamanın standart kurulumu yapılabilir. “Setup Checklist” denilen bu ana ekranda yaptığımız veya eksik bıraktığımız adımları görüp istediğimiz adıma rasgele atlayabiliriz. Listede içinde sağa doğru yönlendirilmiş ok bulunan düğmelere tıklanarak alt menülere ulaşılır. Örnek olarak Okul Ders Programı projesinin kurulum aşamaları anlatılacaktır. Her adımın işlendiğini seçeneklerin yanında beliren kırmızı bir check işareti ile anlayabiliriz.



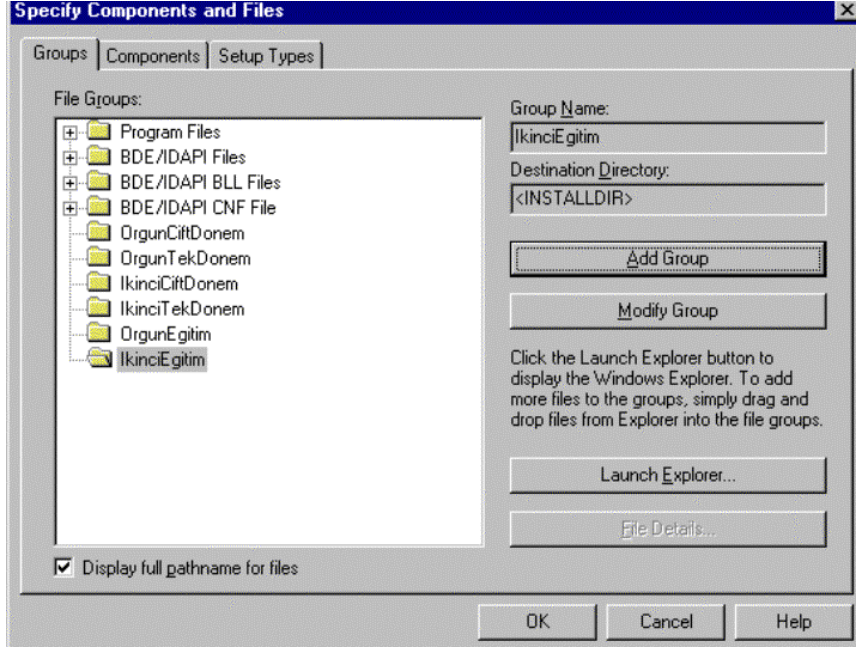
Resim 2 – Görsel tasarım penceresi

İlk olarak kurulumda genel olarak kullanılan uygulama adı, versiyon, kurulum penceresinin rengi kurulum penceresinde yer alacak yazı ve resimler, programın kurulacağı varsayılan hedef dizin belirlenir. Kurulum bitince çalıştırılabilecek program da "Application Executable"'e Browse – Gözet düğmesine basılarak belirtilir.



Resim 3 – Veritabanı düzenlemesi

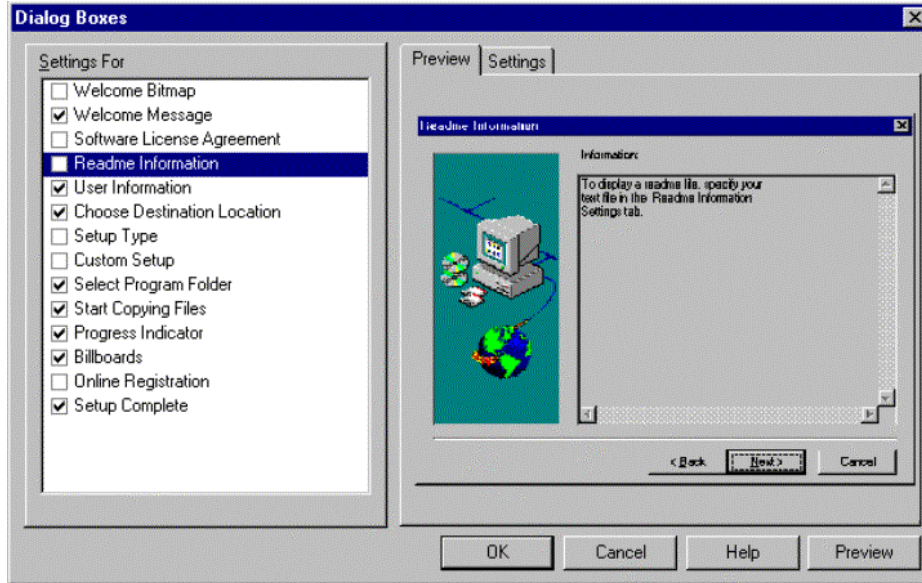
İkinci adımda program eğer veritabanı kullanacak ise Borland Database Engine (Borland Veritabanı Motoru) işaretlenip programda kullanılan alias (takma isim, veritabanlarının bulunduğu dizininin takma adı) "Settings" düğmesi ile ayarlanır.



Resim 4 – Kurulum tipi ve dosyaların ayarlanması

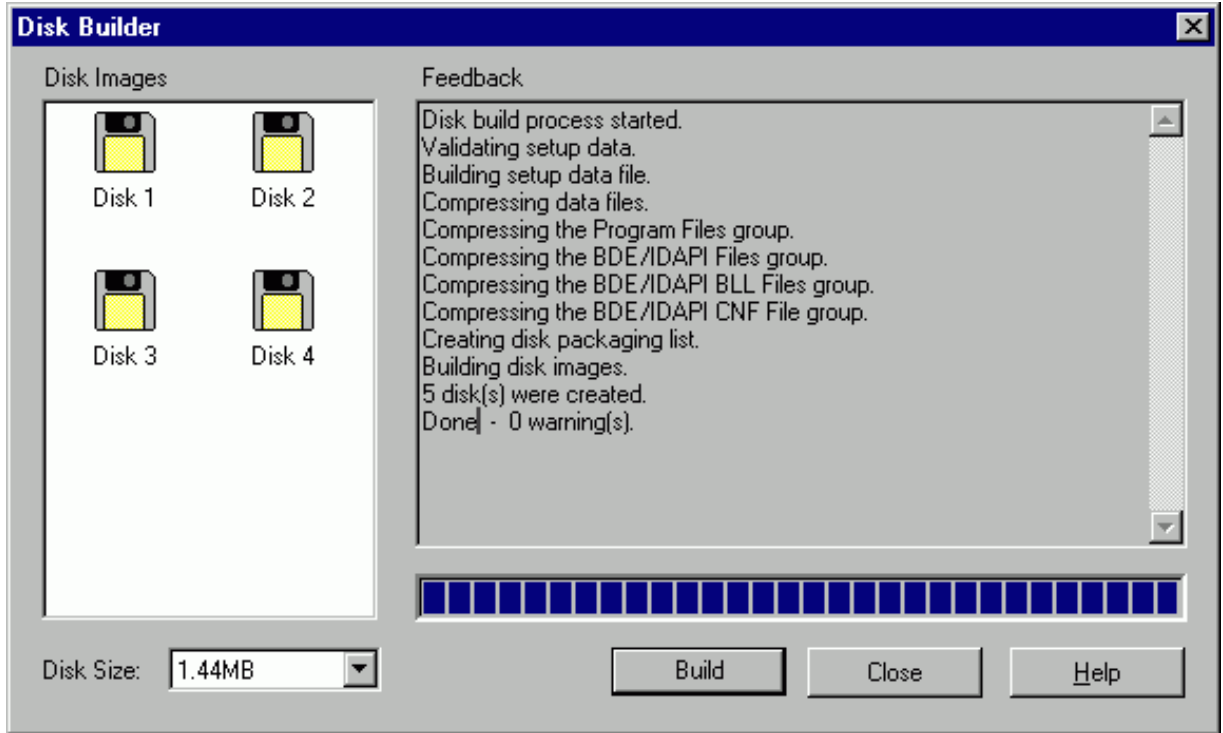
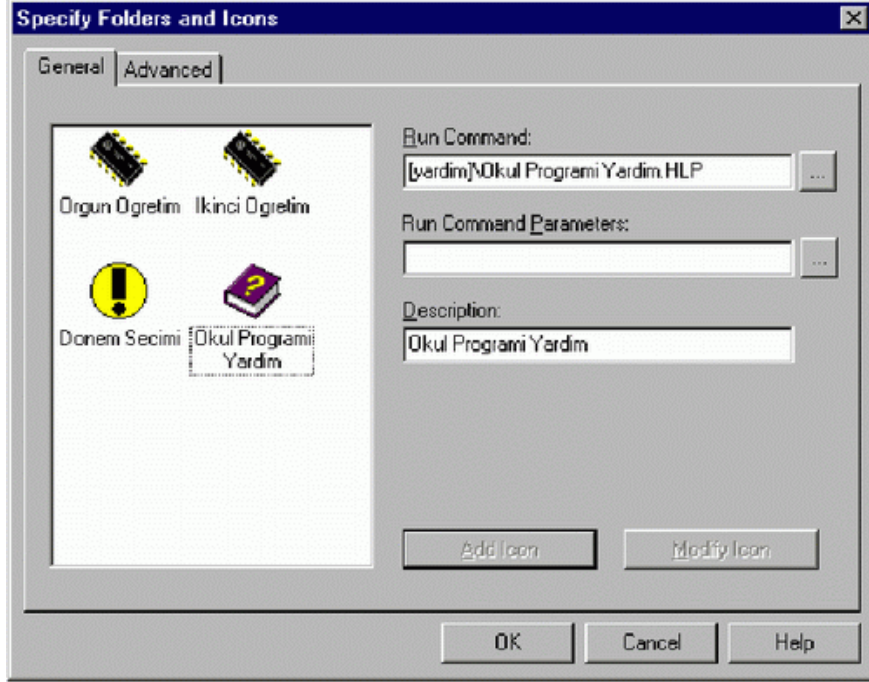
Veritabanlarının kurulacak dizine göre de aliasları ayarlandıktan sonraki önemli adım InstallShield'in hangi programları sıkıştıracağı ve hangi dosyaları hangi dizine kuracağını ayarlandığı "Specify Components and Files" seçeneğidir. Bu pencerenin kullanımı çok kolaydır. "Launch Explorer..." düğmesi tıklanarak dosya veya dosyalar sürükleyip bırak yöntemi ile "File Groups"a aktarılır. "File Groups"un amacı kurulumdaki dosyaların kategorize edilip belli kurulum tiplerinin belli dosyaları kurmasını sağlanmasıdır. Yani "Typical" kurulumda sadece örgün eğitimin programı ve BDE (Borland Database Engine), "Compact" kurulumda sadece örgün eğitim programının kopyalanması sağlanır.

Kurulumu hazırlanan dosyalar aslında birçok uygulama ve yardımcı programdan oluşabilir. Bizim projemizde veritabanları, BDE, 3 adet uygulama programı ve bir yardım dosyası bulunuyordu. Bu dosyalar daha düzenli olması bakımından "Add Group" düğmesi ile gruplanır ve "Destination Directory"leri ayarlanır. <INSTALLDIR> bir değişkendir ve varsayılan değeri bizim program için "<Program Files>\Okul Ders Programı" dizinidir. <WINDIR> ve <COMMONFILES DIR> da kullanılabilen diğer değişken dizin seçenekleridir.



Resim 5 – Kurulum Diyalog kutuları belirlenmesi

Son adımlara yaklaşırken kurulumda bize yol gösteren diğer menülerin seçimi ve ayarlarının yapıldığı "Dialog Boxes" menüsüne gelinir. Hoş geldiniz ekranı, hoş geldiniz mesajı, okubeni bilgisi, kullanıcı bilgisi, kurulum tipi, program dizini seçme, işlem durumu göstergesi, kurulum bitti mesajı ayarları yapılır.



Resim 6 – Kısayolların belirlenmesi ve son aşama olarak disket veya CD hazırlanması

Daha da programlar kopyalandıktan sonra uygulama dosyalarına "Başlat menüsü"ne kısayollarının belirtilmesine sıra gelir. "Run command"ın yanındaki içinde üç nokta bulunan düğmeye basıp ".exe" dosyayı belirtip "Add Icon" düğmesine basılır. "Advanced" kısımdan programlara simgeler belirtilir. Son adım olarak disketlerin oluşturması gelir. "Build" denilince geri bildirim penceresinden hata mesajları ve işlem durumu görülebilir. İşlem bitince "Test Run" ve "Copy to Floppy" komutları verilebilir. "Test Run" ile örnek bir kurulum işlemi yapılır. "Copy to Floppy" ile kurulum disketleri kopyalanarak hazırlanır.

Ek 5 – CD’de Neler Var?

➤ CD içeriği

- ❖ **Dokuman** (284MB) - Ekstralar, tutorial (öğretici), Delphi ve VB için component yani nesneler, ve bir çok minik program
- ❖ **Programlama** (236MB) - Shareware veya freeware olarak dağıtılan derleyiciler
- ❖ **Utility** (0MB) - Sık kullandığım faydalı shareware veya freeware programlar.
- ❖ **Ornekler** (136MB) - Assembler, Basic, C, Delphi, VB, Pascal ve diğer program örnekleri (ActiveX, VRML, PLC, MSeExcel)
- ❖ **Yardım** – CD içeriğini belirten *.html dosyalara erişebilirsiniz. Tüm dosyaların hepsine buradan link veremedim, Windows Gezgin’inden incelemenizi tavsiye ederim.

➤ Diğer

Component : Delphi ve Visual Basic için basit birkaç nesne paketi.

ActiveSkin_VB : 30 gün kısıtlı, formunuza güzel skin (deri) kaplamak isterseniz.

DelphiX : Delphi 3, 4 ve 5 için DirectX kullanımı. DirectSound, DirectPlay gibi DirectX öğeleri elinizin altında.

GLScene : OpenGL destekli oyun veya endüstriyel programlar oluşturabilirsiniz. glscene.org sitesine giderseniz orada bir Türk arkadaşında ürününü görebilirsiniz.

Shape_Delphi4 : Organic Shape de demo sürüm. Yani rast gele zamanlarda programınız çalışırken uyarılar veriyor. Bu nesneler ile ActiveSkin gibi formunuza *.bmp resimleri deri olarak kaplayabilirsiniz.

WavMix : Bu sürümü sadece Delphi 3 ile çalışıyor. Aynı anda birden fazla (en fazla 8 kanal) ses çalabilmeniz için bunu kullanabilirsiniz.

ZPH211 : Bedava, Hint'lerinizi güzelleştirin.

Extras : JavaScript, ipuçları, utility (yardımcı dosyalar)...

JDK : Java Development Kit. Java derleyicisi. İçindeki örnekler hoşunuza gidebilir.

Merlin, Genie, Peedy ve Robby : Microsoft Agent örnekleri. Sisteminizde **MSAgent** kurulu olmalı.

cdebnk23.zip, davetips.zip, ttricks.zip : Tips&Tricks mantığı ile programcılık ve Windows ile ilgili ipuçları.

ANIMENU.ZIP, JAVA sAAT.zip, JavaMENU97.ZIP, JavaOpenGL.zip, javascript.zip, MENU.ZIP, NEON.ZIP, SUPANO.ZIP : Java örnekleri.

ica3.exe, isb30.zip, isa10.zip, cortvrml.exe, vrml10conv.exe, spazz_v24_full.exe, VRML.zip : VRML editörlerinin ve yardımcı programlarının demoları.

MEDIA_dbase.zip, ogrkay_dbase.zip : dbase'de yaptığım örnekler.

MEMTURBO.EXE, mphexer.exe, pn.zip, Recover4all 1.03.zip, tinypersfirewal.exe, UM270.EXE, winkey.exe, mop.zip : Diğer sevdiğim shareware / freeware programlar.

3d_rotat.zip, astar[1].zip, cpu[1].zip, ExitWindows[1].zip, dis86.txt, MakeActor1_3.exe, OktSysInfo[1].zip, sprite.doc, SuperForm[1].zip, XXEMF200en.exe : Programlama örnekleri

FAQ : GLScene ve Delphi3d sık sorulan soruları.

Tutorial : VB, C, C++, Flash, Excel öğreticileri. Linux-HOWTO dökümanları.

➤ Programlama

Gün kısıtlı veya freeware olarak deneyebileceğiniz programlama dillerini burada bulabilirsiniz.

AnfyJava143 : İçindeki hazır JavaScript'leri özelleştirerek sitenize ekleyebilirsiniz. Linkler internet üzerinde çalışmıyor, demo yani.

Applet FloatingMenu Builder : Sitenize güzel menüler yapmak isterseniz.

Asm : Turbo Assembler ile *.asm uzantılı dosyalarınızı *.exe veya *.com yapabilirsiniz.

c-- : c++'a benzer bir dil. Örnekleri güzel.

Delphi5Trial : Delphi5'in 60 günlük deneme sürümü. İnternet'ten kayıtlı kullanıcı olarak ancak kurabilirsiniz. Hem de yayım süresi geçebilir. (expire yani) Kurulum zahmetli.

Dev-C++ : Bedava c++ editör ve derleyicisi.

Dev-Pascal1.2 : Bedava Pascal editör ve derleyicisi.

Genesis3D2 : 3D etkileşimli dünya oluşturma.

Help : **MSHelpWorkshop, OasisSE, chm_help.exe** ve **helppad.zip** kendi *.hlp / *.chm dosyalarınızı yapmanız için.

HelpPC : **HELPPC.COM** ve **TH.EXE** ile assembler, interrupt - kesmeler, PC donanım bilgilerini edinebilirsiniz. Çalıştırmak için klasörlerine gidin.

ipascal : Irie Pascal 2.01 ile *.pas dosyalarınızı edit edip, derleyebilirsiniz. Ama doğrudan olarak *.pas değil *.ipj proje dosyalarını derleyebiliyor.

JavaDecompiler : Bu program ile *.class uzantılı dosyaları edit edebilir çalıştırabilirsiniz. **JDK**'yı da kursanız iyi olur.
VB5CCEIN.EXE : VB5'in bu özel sürümü ile ActiveX veya standart EXE projeleri oluşturabilirsiniz. Aslında Application Setup Wizard kısayolu ile ActiveX Control içeren projelerinizi internete yayınlamanızı sağlıyor. Yani EXE oluşturmuyor.

➤ Utility

Bu klasördeki tüm programlar shareware / freeware olan ve benim sık kullandığım programlardır.

ACDSee30 : Resim izletici ve extraları da (HTML Album Generator, Photo Enhancer...) olan program

Cyblox : 3D editör

Divx4 : Artık standart bir görüntü sıkıştırması (MPEG4)

FerretSoft : WebFerret3.5001 ile internette arama yapılır - freeware

Gimp1.1 : Linux için üretilmiş, MSPaint'in gelişmiş hali

ISXpress1.1 : Delphi programlarınızı Setup haline getirir

Kashmir : 3D editör

Macromedia : **Flash5** ve **Dreamweaver4**'in 30 gün kısıtlı malı ürünleri

MORE : Tweak programı - freeware

MPower : Sabit disk bilgilerini gösterir - freeware

MultiOS : Boot manager - freeware

omentree : JavaScript ile yapılan menü

Regclean : Registry temizleyici - freeware

Sensiva2.51 : Hoş bir fare kullanımı - freeware

Terragen0.8 : Kendi terrain'lerinizi(harita) üretin

Test CPU : Delphi ile yapılmış güzel bir uygulama - freeware

truespace : Truespace3 eski ama çok hoş bir 3D editör - freeware

winrar203 : RAR

Winzip8 : Zip

WWEB1.51 : İngilizce-İngilizce sözlük - freeware

➤ Kitapta İşlenen Diğer Örnekler

meraba.asm : İlk assembler örneğimiz

MAKRO.C : C dilinde basitce makro kullanımı

union.c : union tipli değişken oluşturmak

unionreg.c : union tipli değişken kullanarak sürücü bilgileri listeleme

led : C dilinde paralel porttan bilgi okumak ve yollamak

speaker : C dilinde hoparlörden bip sesi çıkarmak

speaker2 : C dilinde hoparlörden daha güzel bip sesi çıkarmak

speaker3 : C dilinde hoparlörden *.wav dosya çalmak

backproc : Delphi'de uzun işlemlerin nasıl yapılabileceği örneği

create : Delphi'de nesne oluşturma ve yok etme

doner : Yuvarlak form içinde saat örneği

dosya : Writeln ve Readln ile dosyalama

gezgen : Metin, web sayfası, film, ses ve resim dosyalarını gösteren örnek

sql : SQL komutlarını deneyebileceğiniz örnek

surukle : Sürükle bırak ile rapor oluşturma

sysmenu : Sistem menüsüne kendi elemanınızı ekleme

tictoc : TicTocToe oyunu

tusbasimi : Fare ve klavye tuş basımları

dosyalistesi.xls : Excel'de recursive kullanım ile dosya listesi çıkarma

toplama.vbs : Windows Komut Dosyası (WSH) kullanımı

kopyala.bat : DOS'da komut otomasyonuna örnek

sendkeys : VB dilinde tuş basımı olayları

➤ Örnekler Dizinindekiler

➤ Assembler

Assembler (mikroişlemci için yazılan en düşük seviyeli dil) ile hazırlanmış örnek programların kodları:

Diger - Burada birçok kısa ama işe yarar kod var.

Dir - Norton Commander'a benzer dosya yönetim programı

Drivers - Sürücü listesi

editor - *.txt yazım editörü

fridge - Güzel bir grafik efekti
RAPOR - DOS hafıza miktarını gösterir
TEXT - *.txt dosyaları *.exe'ye çeviren çevirir
TSRCLOCK - TSR olarak devamlı çalışan saat

➤ Basic

GWBasic ve QuickBasic dillerin ile hazırlanmış örnek programların kodları:

Kök dizinde grafik ile ilgili birkaç örnek var.

QBasic\Examples dizininde genel amaçlı program örnekleri var.

GWBasic ile 80'li yıllara geri dönüyoruz. "system" komutu ile GWBasic'den çıkabilirsiniz. Dosya yüklemek için F3 tuşu ile load"+dosya adı denir. Çalıştırmak için F2 tuşu...

➤ C Dili

C Dilinde hazırlanmış örnek programların kodları:

Dizinsay : Recursive fonksiyon kullanımı

okul : Okulda yazdığım birçok basit program

TSR2 : Terminate and Stay Resident programcılık

WordText : Gelişmiş bir C projesi.

➤ Delphi

Delphi Dilinde hazırlanmış örnek programların kodları:

3Dfx - Voodoo ekran kartı olanlar için basit grafik efekti

Agent - Microsoft Agent'larının kullanımına basit bir örnek

Atlar - At yarışı ile ilgili veritabanı programcılığı (alias : atdata)

CD - Görev çubuğunda CD takma / çıkarma yapabiliyor

Dergilerim - Bilgisayar dergilerim için yaptığım veritabanı programı (alias : dergiler)

DLL - Şifre ("B") kontrolü yapan program

Dolphin - **WavMix** nesnesi ile tasarladığım hoş bir müzik programı

Glared Test Bouncer - **GLScene** ile hazırlanmış oda içinde zıplayan top örneği

Meraba - Bir klasördeki *.bmp ve *.jpg resimleri gösteren ekran koruyucusu

Ofis - Bir petrol ofisi için hazırladığım veritabanı programı (alias : sonmez)

ParalelPort - Paralel porttaki ledleri yakıp söndürebilirsiniz

qcreator - Query Creator ile bir alias içindeki veritabanılarını görüp inceleyebilirsiniz

Shape - OrganicShape kullanarak yaptığım basit bir skin programı

Soru - Sınav sorusu programı

sozluk - İngilizce -Türkçe sözlük programı (alias : sozluk)

surukle - Sürükle bırak ile QuickReport kullanımı

TemplProto - Herhangi bir program yaparken yardımcı olabilecek program kalıbı

Tercuman - İngilizce'den Türkçe'ye çeviri yapmaya çalıştım ama yarım kaldı (alias : tercuman)

Thread - İşlemlerinizi kanallara bölerek daha hızlı çalışmalarını sağlayabilirsiniz

➤ Diğer Diller

Diğer dillerde hazırlanmış örnek programların kodları:

ActiveX - Internet sayfası içinde çalışan program

Excel - Basit birkaç tane makro örneği, **YAZIYLA.XLS** , **tarih_goster.XLS** ve **userform.xls**.

PLC - Programmable Logic Control, okulumuzun ürettiği **simülatörü** ve **slide** (Page up/dn ile ilerlenir) göstericisi.

"Ornekler\Diger\PLC" dizininden istediğiniz *.bat dosyayı çalıştırın.

VRML - "Diger\Extras" dizinindeki programlarla siz de kendi üç boyutlu dünyanızı oluşturabilirsiniz. Örnek **evim.wrl**.

➤ Pascal

Pascal Dilinde hazırlanmış örnek programların kodları:

Diğer, Graf, Intro - Grafik ile ilgili bir çok minik program

Har_Menu - Menu oluşturma

Okul - Okulda yaptığım ilk deneyimler

Prince3D - Çocuk gerçekten iyi çalışmış, o bir Türk.

ScrSaver - Ekran koruyucusu

SM - Grafik ekrana geçiş kodları

Staj - 1 aylık çalışmam **Ariza** ve arkadaşın program örnekleri.

➤ Visual Basic

Visual Basic dilinde hazırlanmış örnek programların kodları:

BMP_Show - *.bmp resimleri gösterir

Dosyalar - Aslında böyle çalışmamak gerekirdi, ama tüm yaptıklarımı tek dizine kaydetmişim.

Final - Gerçekten final gibi, hesap makinesi çalışmam. Klavye ve fare entegrasyonu iyi.

Fraktal - Ne olduğunu tam anlamadım.

HesMak - Çok basit bir hesap makinesi

LabelSoru - Sınav sorusu hazırlama

Listbox - Arabirimi berbat ama çalışma hiç fena değil. Biraz inceleyin.

MsgBox - Mesaj kutusu üretir

RGB - Red green blue değerleri

Sinav_soru - Soru kayıt programı

SoruForm - Sınav sorularının form hali

Telefon - Telefon kaydı

Winline - 4 aylık, 2001 tarihli çalışmam. Basit bir oyun.

Ek 6 – Sözlük

İsim	Okunuş	Anlam
3D	tri di	üç boyutlu (3B)
access	ekses	erişmek
administrator	ıdministreytır	idareci
agent	eycınt	ajan, memur
allocate	elokeyt	atama, yer bulmak
application	eplikeysın	uygulama
area	eria	alan
argument	argyumınt	parametre, argüman, fonksiyona veri gönderme
artificial	artifişıl	yapay
assemble	ısseml	toplamak, bir araya getirmek
associate	essoşieyt	ilişkilendirmek
attribute	ettribyut	öznitelik, özellik
backup	bekap	yedekleme
base	beyz	taban, merkez, baz
batch	beç	toplu dosya
BIOS	bayos	bilgisayarın temel donanım bilgilerini tutan çip
binary	baynırı	ikilik sayı sistemi
boolean	buulıin	mantıksal değer
boot	buut	bilgisayarın açılışı (bootup)
branch	branç	programın başka yöne dallanması
buffer	bafır	ön bellek, tampon, ara hafıza
bug	bag	böcek, hata
C++	si plas plas	C dilinin üst versiyonu
cache	keyç	ön bellek, kaşe
caret	karıt	^ işareti, şapka
cartridge	kartric	kartuş
cascaded	keskeydid	basamaklı, azalan
CGI	si ci ay	internette veritabanı uygulamaları
chain	çeyn	zincir
char	çar	tek karakterlik değer (character – keriktır)
circuit	sörkit	devre
cluster	klastr	küme, her track – iz'deki sektör grubu
code	kod	kod, programda kullanılan komut kümesi
collapse	kıleps	göçmek, gizlemek
command	kımand	komut
compare	kimpeyr	karşılaştırmak, comparison – karşılaştırma
component	kimpounınt	nesne, bileşik
conditional	kındışınıl	şart, koşullu
conflict	kınflikt	çakışma
console	konsoul	ekran ve klayve arabirimi (CON)
convert	kınvört	çevirmek, biçim değiştirmek
current	körrınt	geçerli, şu anki
cursor	körsır	imleç, fare oku (mouse pointer)
database	deytabeyz	veri tabanı
debug	dıbag	Windows'un böcek ayıklama programı

default	difolt	varsayılan
Delphi	delfi	Yunanistan'da Parnassus Dağı eteklerinde eski bir şehir
demonstration	demonstreysın	reklam, tanıtım, demo
design	dizayn	tasarım, dizayn
destination	destineysın	hedef
device	divays	alet, aygıt
diagnostic	dayagnostik	sebebi bilinmeyen bir hatayı giderme, diyagnoz
dialog	dayalog	diyalog, söyleşme
dimension	daymensın	boyut
disassemble	disesseembl	parçalamak
dot	dot	nokta, ekranın fiziksel en küçük ışık noktası (dpi <i>di pi ay</i> – dots per inch)
draft	draft	taslak
drag and drop	dreg end drop	fare ile sürükle ve bırak işlemi
driver	drayvır	sürücü
exceed	eksiid	aşmak
entry	entri	giriş (entrance)
environment	envayırmint	ortam, çevre
evaluation	evaluateysın	değerlendirme
expanded	ekspendid	genişletilmiş
explorer	eksplorır	gezgin
extension	ekstenşın	uzantı
fail	feyl	hata, failure – başarısızlık
false	fols	mantıksal yanlış
FAQ	ef ey kyu	Frequently Asked Questions – Sık Sorulan Sorular
FAT	fet	File Allocation Table – Dosya Atama Tablosu
feature	fyuçır	özellik, aksam
file	fayl	dosya
float	flout	kayar noktalı gerçel sayılar
frag	freg	parça (fragment , defrag - birleştirmek)
frame	freym	çerçeve, iskelet, kare
function	fankşın	görev, işlev, fonksiyon
GUI	ci yu ay	Graphical User Interface – grafiksel kullanıcı arabirimi
guide	gayd	önder, kılavuz
handle	hendl	tutamaç, kulp, tanıtıcı
handshake	hendşeyk	el sıkışma, anlaşma
hexadecimal	heksadesimıl	onaltılık sayı sistemi
HTML	eyç ti em el	Web sayfalarının temelini oluşturan dil
IPX	ay pi eks	Internet packet exchange, başka bir internet / intranet bağlantı yapısı
icon	aykın	simge
identifier	aydentifayır	tanımlayıcı
image	imic	resim, imaj
inch	inç	2.54 santimetre
information	infomeysın	bilgi, enformasyon
initiation	inişieysın	ilkeme, kullanıcı ayarlarının saklandığı dosyalar (*.ini)
install	instol	kurmak, installation – kurulum
instruction	instrakşın	komut
instrument	instrumınt	enstrüman, araç
integer	intecır	tamsayı
introduction	introdakşın	tanıtım, intro

item	aytım	nesne, öge
java	cevi	Sun Microsystems tarafından çıkartılmış, platformdan bağımsız programlama dili
kernel	kernıl	çekirdek
log	log	kayıt tutmak, döküm yapmak
logical	locıkl	mantıksal, boolean
match	meç	uymak, oyun, birbirini tutmak
maximize	maksimayz	en büyük yapmak, ekranı kapla
message	mesic	mesaj, ileti
minimize	minimayz	en küçük yapmak, simge durumuna küçült
missing	mising	eksik, kaçırmak
multimedia	maltimedya	çoklu ortam
network	netwörk	ağ
object	ibcekt	nesne
OCR	ou si ar	Optical Character Recognition – Optik karakter tanıma, yazılımı tarayıcıların yanında verilir
OpenGL	oupın ci el	Özellikle VC gibi bir dili kullanarak grafik kartını programlama
optimize	optimayz	en iyi hale getirmek, optimize
optional	opşınıl	isteğe bağlı, opsiyonel
page	peyç	sayfa
parameter	perimitir	parametre, argüman, anahtar
parity	periti	parite, eşlik
partition	partişin	disk bölümü (c: gibi)
path	pet	yol, izlek, patika
pattern	petırn	doku, dizi
PCMCIA	pi si em si ay ey	Özellikle not defteri bilgisayarlara takılan dış kart veriyolları, PC kartı
perform	pörform	gerçekleştirmek
peripheral	perifırl	çevresel birim
pin	pin	giriş çıkış için kullanılan port – veriyolu iğnesi
pixel	pıksıl	Ekrandaki yazılımla elde edilebilen nokta (picture element/picture cell)
pointer	pointır	işaretçi, fare imleci
port	port	kapı, fiziksel iletişim hattı
present	prezınt	var olmak
previous	priviis	önceki
primary	praymırı	birincil
priority	prayoriti	işlemciye yaptırılan işlemlerin öncelik değerleri
procedure	pirisiicır	yordam, prosedür
project	pricekt	proje, tasarı
prompt	prompt	uyarı, MS-DOS sistem hazır simgesi c:\> gibi
protect	prıtekt	korumak, protection – koruma
queue	kyu	kuyruk
real	riil	kayar noktalı gerçel sayılar
recognize	rikıgnayz	tanımak
record	rikırd	rekor, kayıt
recover	rikavır	bilgileri kurtarmak
recursive	rikörsiv	kendini çağıran fonksiyon
register	recıstır	yazmaç, kayıtlı kullanıcı olmak
registry	recıstri	Windows kayıt bilgileri

reserve	rizörv	ayırmak, rezervasyon
revert	rivört	geri dönüştürmek, kurtarmak
root	ruut	kök, ana dizin
scene	sin	sahne
script	sıkript	yardımcı programlama dili, el yazısı
scroll	skrol	kaydırmak
SCSI	sıkazi	Seri IDE bağlantıya göre daha gelişmiş bağlantı türü (Small Computer System Interface)
sector	sektir	bölge, disk iz parçası
sequence	sikuins	sıra
server	sörvir	ana bilgisayar, sunucu
set	set	küme, takım, ayarlamak (setting)
sign	sayn	işaret, signature – imza
simulate	simyuleyt	benzetmek (simulation - benzetim)
slot	slot	yuva, kart takılan yarıklar
specify	spesifay	tanımlamak, belirtmek
speech	spiç	konuşma
sprite	sprayt	grafik programcılığında kullanılan hareketli nesneler
SQL	es kyu el	Structured Query Language – Yapısal Sorgulama Dili
stack	stek	yiğın, geçici bellek alanı
status	steytıs	durum; statement – deyim, komut
structure	strakçır	yapı
subdirectory	sabdayrektöri	alt dizin, klasör
substitute	sabstityut	başkasının yerine koymak
success	sıkses	başarı
supervisor	supıvayzır	yetkili
support	sıport	destek
surface	söfis	yüzey, görünüş
suspend	sıspend	askıya almak
switch	sviç	anahtar, yer değiştirmek
TCP/IP	ti si pi ay pi	İnternetin çalışmasını sağlayan alt yapı (Transmission Control Protocol / Internet Protocol)
terminal	törminıl	son, uçbirim
thread	tred	kanal
toggle	togl	iki şeyin birbirleri ile yer değiştirmeleri (swap)
track	trek	iz, fiziksel disk dairesi
true	tru	mantıksal doğru
truetype	trutayp	boyutu değiştirilebilir yazı tipi
try	tray	denemek
tutorial	tyutrıl	öğretici
unit	yunit	kod yazılan alan
utility	yutiliti	yardımcı programlar
variable	verayıbl	değişken
virus	vayrıs	virus
volume	volyum	sürücü
warning	worning	uyarı
watch	voç	çalışma anında değişkenleri seyretmek

Dizin**#**

#include	41
----------	----

\$

\$HTTP_COOKIE_VARS	290
\$HTTP_GET_VARS	290
\$HTTP_POST_VARS	290
\$HTTP_SESSION_VARS	290

* *	155
*.3ds	164
*.acs	69
*.asm	13, 333
*.asp	257
*.aspx	270
*.avi	43, 146
*.bas	27, 335
*.bat	241, 243
*.bmp	93, 168
*.bsp	165
*.c	40
*.cgi	206
*.class	348
*.com	18
*.cpl	246
*.csv	80
*.db	84
*.dbf	84, 323
*.dll	89
*.dmf	189
*.dpk	164
*.dsk	80
*.exe	18, 68
*.fla	282
*.gif	146
*.hex	332
*.htm	247, 253
*.inf	80
*.ini	80
*.iwz	343
*.jal	334
*.jp*g	93
*.js	240
*.map	165
*.md2	164
*.mdb	279, 326
*.mov	43, 146
*.mp3	146

*.mpg	146
*.msc	80
*.obj	18, 28
*.ocx	231
*.opt	80
*.pas	64
*.php	281
*.pl	257
*.prg	323
*.proje	272
*.rc	68
*.res	67, 272
*.rpm	289
*.scr	93
*.swf	253
*.tga	146
*.tlb	229
*.tpu	41, 64
*.txt	20, 223, 247
*.vbs	240
*.vxd	110
*.wav	109
*.wrl	164, 313
*.x	164
*.zip	164, 313

.

.NET	270, 297
------	----------

?

? (parametre)	13
? (print)	63

^

^ operatörü	64
-------------	----

+

++	29
----	----

1

16f84	332
16f877	332

3

3d	312
3d sohbet	312
3Dfx	164
3DMax	313
3DSMax	164

6

68000	246
-------	-----

7

7447	282
------	-----

8

80 portu	327
800x600	182

A

AbriaMerlin	279
Access	62
Action Script (Flash)	253
Action Script 2.0	317
Active Server Pages	257
ActiveForm	230
ActiveSkin	347
ActiveX	229, 247
açılış disketi	245
ADC devresi	110
Add all fields	86
ADSL	327
AGP	246
Akbil	107
akış diyagramı	339
alfa nümerik	102
algoritma	182, 339
alışveriş	314
alias	84
Alpha	246
Alt tuşu	217, 342
AMD	246
Amiga	246
analog	110
and	10
animasyon	73, 313, 317
anket	257
antialiasing	313
Apache	206, 279, 289
API	110
App	66
Apple	246
application	120

Application nesnesi	66
arama	36
Artificial Intelligence	5, 160
asansör	214
ASCII	341
asm	42
ASP	326, 364
ASP.NET	297
ASPack	272
assembler	332
Assembler	10
Atari	246
auto	31
AutoCAD	43, 240
autoexec.bat	245
AutoLISP	240
AX	11

B

Baby Web Server	326
Backspace tuşu	218
base	105
Basic	26
batch	241
bayrak	15
bayram günleri	238
BDE	62
BDE Administrator	84
Benchmark	273
BeOS	7, 246
BGI	56
BIOS	10
bi – directional	103
bi – directional bubble sort	35
binary	10
binary (ikilik arama)	36
bitmap	43, 67
bitsel işlemler	34
BlueDream	274, 276
BlueDream projesi	270
Bluetooth	326
boolean	30
boot	10
Borland Graphic Interface	56
Borland Pascal	55
böcekler	144
brcc32.exe	67, 272
break	65
bubble sort	35
buffer	103
Build with runtime packages	62
bump mapping	43
Button	81
BX	11
ByteBool	65

C

C dili	29
--------	----

C++	7, 29, 46
CAB file	230
Cache	11
calculated field	209
call	19
canvas	64, 114
Caps Lock	73
Carriage Return	19, 342
carry	15
case	44
case sensitive	30, 340
Casio FX-880P	7
CD	364
Centronics	106
CGI	206, 257, 364
char	30
Character	252
Chart	176
ChartFX	176
chmod	290
Chr(ascii_deger)	102
CISC	11
collision	164
ColorDialog1	117
COM	241
COM port	107
Combobox	84, 120
Command1(Index).Caption	222
Command16.SetFocus	222
CommandButton1	236
compile	62
compiler	28
component	83
Components	141
config.sys	245
const	31
Control Panel	245
Control tuşu	217
Corel	240
CorelDraw	43
CorelScript	240
Cortona	312
cos	114
Counter-Strike	273
crack	364
Crimson Editor	279
cross kablo	107
crt ünitesi	64
Ctrl+Enter	253
CX	11
Cyrix	246



çerez	208, 290
çok işli	246
çok kullanıcı	246
çokgörevlilik	120
çoklu işlemci	246
çoktan seçmeli sorular	290
Çözücü Eklentisi	294

çözünürlük değiştirme	164
-----------------------	-----

D

DAC devresi	110
DanceEJay	189
data komutu	68
data segment	30
dbase	116, 347
dbase 3	323
DBChart	176
DBGrid	84
DBNavigator	116
DC	10
dcc32.exe	274
debug	5, 14
Debug menüsü	63
decode	102
decrypt	102
defrag	246
Delphi	62
Delphi Türk Kod Bankası	144
Delphi1.0	62
DelphiX	164
demo	110
demonstration	110
Denetim Masası	245
denklem	294
Dependency Walker	90
derlemek	67
derleyici	28
Design Mode	237
DHTML	314
Dictionary On-Top	115
dijital	11
dinamik dizi	30
dinamik hafıza kullanımı	33
dinamik IP	327
dir more	55
DirectX	164, 313, 347
disassembler	63
Disassembler	274
div	15
dizi	222
diziler	30
dizin seçme diyalog kutusu	67
DLL - Dynamic Linking Library	89
Dock - gömülme	168
donanım kesmeleri	19
Doom	165
DOS	18
DOS programcılığı	5
dosyalama	223
DotNET	274, 291
Dreamweaver MX 2004	314
durak noktası	63
duvar kağıdı	168
Duyarlılık	296
duyuru	314
düşünmeyi öğretmek	137
DVD	76

DX	11
DXSound	200
dynamic data exchange (DDE)	217
Dynamic Text	252

E

ECP – Extended Parallel Port	106
edit	27
Edit nesnesi	116, 128
ekran koruyucusu	93, 349
eksabyte	17
embed	255
empty()	290
Enable Macros	232
encode	102
Encrypt	102
end.	64
Enter	128
Enterprise	230
environmental mapping	43
eposta	314
EPP – Enhanced Parallel Port	106
EPROM	332
ESC tuşu	48
ethernet	246
Events	62
Excel	80, 232, 236
executable	28
exit	65
Export Table	90
ext	246
ext2	246
ext3	246
extended filesystem	246
extended özelliği	223
extern	31

F

F1	364
false	30, 65
fare	41, 67
fare kullanımı	48
fare tıklamasının	67
FAT	246
fatura	198
fflush(stdin);	33
filtreleme	84, 301
Flash	43, 252, 317, 347
Flash RAM	246
flat	137
flicker	114
Flip – Flop	10
float	30
FloatToStr	102
fly-by evaluation hints	63
fonksiyon	27, 33, 38
font	43
FontDialog1	117

for	44
form	111
Form1.Canvas	114
Format fonksiyonu	65
formu fare ile sürükleme	73
forum	257
fotoğraf makineleri	246
Fourier	294
FoxPro	62
fps	253, 317
FPS	313
FPU	11, 15
frame	253
freeware	347
from	340
FTP	206
full duplex	108

G

game port	110
GDI	89
gdi32.dll	89
GeForce	164
Gelecek Linux 2.0	289
getch()	29
getchar()	29
getche()	29
Getimage	56
gets()	29
getURL();	255
giga	17
Giriş / Çıkış	106
GLScene	164
Google	290
Goto	26
gotoAndPlay (1);	253
görev çubuğu	76
Görev çubuğu	67
görev çubuğunda simge	80
Görsel Nesne Kütüphanesi	62
GPF	89
grafik ekran	48, 56
grafik nesnesi	176
gramer	200
GridLayout	160
grid	160
GUI	62
güvence	66, 152
GWBasic	26, 349

H

haftalık ders programı	179
haftanın günü	44
Half – Life	164
half duplex	108
harita	320
HasColumnHeader	117
hashing (kıyma arama)	36

HelpPC	19
hesap makinesi	58, 218, 350
hesaplanan alanlar	198, 301
hex – editör	14
hexadecimal	10
hipotenüs	114
Histogram	294
Hizmetler (Services)	80
host	257
hourglass	120
HTML	7, 146, 252
httpd.conf	281

I

I/O	106
IBM OS/2	246
IDE	62, 297
IEEE	30
IIS	206, 297
Image	111
Image Editor	67, 137
Imagesize	56
Index	84
InputQuery	66
InstallShield Express	343
InterBase	340
Internet Scene Builder	312
IntraWeb	208
IntToStr	102
ISIS Proteus	282

İ

iconlib.dll	89
identification	255
idle – boş	120
iframe	247
iif deyimi	32
ikinci derece denklem	339
ikincil indeks	84
immediate	63
in deyimi	65
include	281
index	223
infrared	108
İngilizce çeviri	200
initiation	80
inside – out activation	229
int değişkeni	30
intel	11, 246
interpreter	28
interrupt	19, 347
intro	110
introduction	110
iODBC	279
istatistik	294
işar	198
iterative (tekrarlan)	36

J

JAL(Just another language)	334
Java	62, 364
Java Development Kit	160
JavaScript	69, 247, 252, 279, 282, 298, 314, 347, 364
joker karakterleri	242
joystick	110
J-script	240

K

kalem ile çizim	213
kanal	120
Karakter animasyonu	313
Kate metin editörü	289
kavşak	317
kayan yazı	65, 247
kayan yıldızlar	164
kaydırma	211
kelimeyi aramak	256
kendi yazı tipiniz	48
kernel interface	246
kernel modu	246
kernel32.dll	89
kesirli sayıları	30
Key	84
keyframe	253
Kızılötesi	108
kilo	17
kilobyte	242
klavye	42
Konqueror	289
konsol uygulamaları	206
kullanıcı modu	246
küme işlemleri	65
Kylix	206

L

Labcenter ISIS Proteus	332
label	222
Label	117, 128
Label1	155
ladder diagram	10
layer	253
Layer	43
led – light-emitting diode	103
Left\$()	64
legal	364
LILO	246
Lightwave	164
Like	84
Line Feed	19
linear (doğrusal arama)	36
Linux	29, 246
LoadFromFile	152
logo	43
long	30

LongBool	65
Lookup field	86
loop	253
LPT	103, 106

M

Machintosh	246
Macromedia Flash	164, 252
Macromedia FlashMX 2004	274, 282
Magneto Optik Disk	246
main()	29
makro	38
Mandrake 9.0	289
manuel	245
masaüstünün resmi	213
maskeleme	106, 211
Masm	18
math uniti	64
matris	60
MaxScript	240
Mayın Oyunu	141
mbr	246
md5	281
mega	17
mem	18
Memo	152
memory	218
MemoX	270
menü	323
metin kutusu	48
MIPS	246
micro	17
Microchip	332
microcontroller	10
Microsoft Access	37, 257, 281, 323
Microsoft Agent	69, 347
Microsoft Development Environment 6.0	248
Microsoft Visual FoxPro	322, 323, 327
Microsoft Visual Studio	248
Microsoft Windows	4
Mid\$()	64
midas.dll	272
mili	17
mirror set	246
misafir defteri	257
MMX	11
MOD	246
Modem	246
Module	233
modül	232
modüler	106
moricons.dll	89
Motion Tween	253
mouse	40
mov	19
movieClip	317
Mozilla	279, 289
MPEG4	348
MPLAB	332
MS SDK Help Files	89

MSDN	69
MSDN Library	73
MS-DOS	244
msdos.sys	245
MSSQL Server	298
mul	15
multi tasking	246
multi thread	120
multi user	246
Multiline	252
MultiSelect	223
multiservis	279
Multitasking	120
MySQL	279, 280, 285, 340

N

nano	17
nanosaniye	246
nesne oluşturmak	81
Netscape	240, 279
Norton Commander	348
Not girişi	209
notepad	4, 27
Novell Netware	10
NTFS	246
Null Modem kablosu	107
Num Lock	73

O

OasisSE	189
Object Inspector	62
Object Oriented Programming	47
OCX	229
ODBC	62, 279
Office 2000	69, 236
Office 97	69, 236
Office Asistanını programlamak	75
Office XP	69
OLE	241
OLE Automation	229
OmniHTTP	279
OnIdle	121
op-code	19
OpenGL	164, 246, 271, 313, 347, 364
Opera	279
Oracle	340
Ord(tek_harf)	102
oturum	289
overflow	15

Ö

özel karakterler	342
------------------	-----

P

Paint	168
PaintBrush	168
PalmPC	297
palmtop	270
paralel port	326
parametre	13, 93, 120, 244
parametrik	110
parity	15
Pascal	347
Pascal üçgeni	207
patch	89
path	84, 243
PC	246
Peek	68
Pentium	4
perabyte	17
Personal Web Server	257
Photoshop	43, 164
PHP	282, 285, 314
PHPConfig	281
phpMyAdmin	289, 314
PHPTriad	279, 289
PIC	332
PICBasic	332
PIT – Programmable Interval Timer	108
Pi sayısı	114
Pic Mate	336
PickList	84, 176
pico	17
pifmgr.dll	89
pin	110
pixel	253
PLA	10
PLC	349
plug-in	164
pointer	31, 65
Poke	68
PopupMenu	80, 117
Port komutu (Pascal)	106
POST	17
potansiyometre	110
pow (taban, üs);	29
power(x,y);	64
PowerPC	11, 246
primary index	84
Prince3D	349
Print Screen	21, 217
printer	60
printf();	39, 44
priority	120
process	120
ProcessMessages	121, 155
programın bir kere çalışması	66
Programmable Logic Controller	10
Projeksiyon	213
Properties	62
prosedür	27, 33, 38
protokol	108
prototip	33

Publish (Flash)	253
Putimage	56
PWS	257

Q

QReport	209
QRExpression	209
QRLabel	117
Quake	164
Query	340
Query1	176
quick sort	35
QuickReport	117, 349
QuickView	90

R

radyal	114
RAID	246
RAM	11, 17, 18
RAMDISK	245
re deklere	33
read komutu	68
Readln	152
real – time	43
Record New Macro	237
recursive	234
Redundant Array of Independent / Inexpensive Disks (RAID)	246
RefEdit.Ctrl	236
register	11, 31
register_globals	281, 289
ren komutu	244
render	43
reset	31
resim yollama	290, 314
resource	67
ResourceHacker	284
Resume	120
ret	19
reverse engineering	63
RGB	110
RISC	11, 332
Right\$()	64
robot	320
ROM	10
root	257
RS-232	107
run time	63
rundll32.exe	90
RXD	107

S

sample	110
Sample	164
sanal makine	246

SaveToFile	152
sayfalama	301
scale	283
scandisk	246
scanf();	33, 44
scroll	165
Scroll Lock	73
SCSI	246
Secondary Index	84
select	340
selection sort	35
Sendkeys	217
sequential	120
seri port	107
session	281
setup	84
SetVariable	255
shareware	347
shell32.dll	89
Shift tuşu	217
shl	15
short	30
ShowInTaskbar	64
shr	15
SHTML	269
sıralama	301
sin	114
sipariş	314
Sistem DSN	281
sistem menüsü	126
sizeof()	31
SMTP	246
SoleBee	270, 274
sözlük	200, 348, 349
speaker	108
speech	73
SpeedBar	62
SpeedButton	137
SPP – Standart Parallel Port	106
SQL	62, 84, 209, 281, 298, 340
SSE	11
SSL	269
stack segment	30
static	31
stok	198
strcmp	30
stripe set	246
StrToFloat	102
StrToInt	64, 102
struct	31, 47
Studio Max	164, 240
Sun Sparc	246
Suspend	120
sürükle	189
sürükle bırak	48
sürükle ve bırak	117
sürüklemek	67
swap	60
Swap	246

S

şifre çözücü	364
şifreleme	102

T

Tab efekti	128
Table (Access)	257
tabulator	48
tag	253
takvim yapmak	238
taskbar	76
Tasm	18
tatil günleri	238
telnet	246
tera	17
terminal	10
Terminate and Stay Resident	46, 349
Terragen	164
Text nesnesi (Flash)	252
Text Options	252
texture	43
thread	120, 200
ticari uygulamalar	314
Time Critical thread	120
time sharing	120
Timer	111
Timer1	80
toggle	63
Tolerans	296
Toolbar	168
ToolBook Instructor	229
ToolBox	69
ToolPak Çözümleyicisi	294
trafik	317
transistör	11
transparency	213
trigonometri	114
true	30, 65
Truespace3	348
truetype font	43
try except güvencesi	66
TSR	46, 80
TTL	103
Turbo Pascal	55
turistik gezi	313
Tutorial	164
Türkçe	64, 116
twip	65
TXD	107
Type Library	229
type oluşturma	31
typecast	65

U

UART	108
UDMA	246

uninstall	246
union	31
unit	64
Unix	29, 246, 257
USB	246
user interrupt	19
user32.dll	89
UserForm	236
uses	56
UTF-8	282
utility	5
uzaktan kontrol	326

Ü

üç boyutlu	43
------------	----

V

var	29, 64
Variable	252
variant	65
VB.NET 2005	301
VB6	326
VBA	69, 364
VBScript	69, 364
VBX	62, 229
VCL	62
vcl50.bpl	62
vektörel	43
veritabanı	257
View Code	236
virtual machine	246
virüs	364
Visual Basic for Applications	231, 232
Visual Basic.NET	297
Visual Studio 6	323
void	29
volatile	31
volume set	246
Voodoo	349
Voodoo Banshee	164
VRML	164
VRML (Virtual Reality Markup Language)	5, 364
VRML2.0	313
VRMLPad	313
VS Developers	297
VXD (Virtual Device Driver)	110

W

WavMix	189
Web Server Application	208
Webdings	137
WebModule	208
where	340
Win 3.1	62, 121
Win32API	89
win386.swp	246
Windows 2000	246
Windows Gezgini	146
Windows Millennium	93
Windows NT	246
Windows programcılığı	5
Windows Scripting Host	5, 240
Windows XP	289
Windows'u kapatma	76
Windows\Command	243
Windows\System	246
WinLines	160
WordBool	65
Wordwrap	252
Writeln (dosyalama)	152
writeln()	64
www.microsoft.com	69
WYSIWYG	43

X

x - y düzlemi	114
X Window	246
XingMPEG Player	240
xorput	56

Y

Yakınsama	296
yapay zeka	320
yazı tipleri	43
yazıcı çıktısı	117
yığın dosyası	241
Yineleme	296
yorumlayıcı	28
Yunus	270, 274

Z

zaman paylaşımli	120
zlportio.sys	272

Son söz

CD'nizi kaybetmeyin. Örneklerimiz CD'den bağımsız da olsa, dikkat ederseniz çoğu yerde şu CD'nin şurasında gibi ifadelerle CD ve kitap organik bir bağ içinde. Genel olarak CD'yi incellerseniz ilgi alanlarımdan grafik (VRML, OpenGL...), işletim sistemleri, yardımcı programlar olduğunu görürsünüz. Müzik yapmak (sadece MP3 dinlerim), resim işleme (Adobe Photoshop), CAD – Computer Aided Design, donanım (robot, PIC) programlama pek ilgimi çekmiyor. Şimdilik kitapta sadece daha iyi bildiğim dillerden örnekler verdim. Diğer deneyimlerime burada yer vermek gereksiz gibi geldi bana.

Bu arada legal – kanuni programlama yapmaya çalıştık. Biliyorsunuz ki bu diller kötü amaçla da kullanılabilir. Virüs, crack veya şifre çözücü gibi şeylerle uğraşmadım, tavsiye de etmem. Özellikle grafik, dosyalama, sistem bilgilerini elde etme, programlama algoritmaları ve problem çözümleri gibi konulara ağırlık verdim.

Sonuç olarak C, Basic ve Pascal'ın hepsinden bir şeyler yapabilir halde olmanız gerekiyor. Birbirine yakın oldukları için, C dilinde Pascal, Delphi dilinde de VB ile ilgili bilgiler verdim. Eğer Basic biliyorsanız ASP, VBScript, VBA gibi diller sizin için sorun olmaz, C dili biliyorsanız CGI (*si ci ay diye okunur*), PHP (*pi eyc pi diye okunur*), JavaScript'te de sorun yaşamazsınız. İlle de son versiyon dillere sahip olmak zorunda değiliz. Uzun süre Delphi 3 kullandım, Delphi 7 ve VB 6 çoğu zaman her işinizi görür. Eğer çok temel değişimler olmamışsa yeni versiyon sadece gelişimin bir uzantısıdır. Burada versiyon numaralarından bahsetmek riskli aslında, çünkü her an yeni versiyonla veya isim değişikliği ile karşılaşabiliyoruz. Örneğin VB.NET gibi...

En büyük yardımcımız F1 tuşu. Çoğu dilde muhakkak burada iyi yardım konuları bulabilirsiniz. Bunun yanında yeni bir programcı isek, ilk yapacağımız şeyler başkalarının yaptığı programları tekrar yazmak, onları incelemektir. İleri programcılığa geçince de artık sizi kimse tutamaz. Kendi stilinizi edinmiş ve bilgisayar size değil siz bilgisayara istediğinizi yapabilir hale gelmişsinizdir. Yani efendi siz, köle bilgisayardır artık. Bundan sonrasında yapabileceğiniz şeyler benim göremediğim, hatalı yaptıklarımı kendi fikirleriniz ile yoğurarak ilerlemeniz.

Beklediğiniz konuları bulamamış olabilirsiniz, ama daha film bitmedi. Belki başka bir kitapta görüşmek üzere. Programcılık kolay meslek değil, ama hevesli iseniz peşinizi bırakmaz! Zaten o her yerde sizi gözetliyor!

Kaynaklar

- ❖ A'dan Z'ye C Kılavuzu – Kaan Aslan
- ❖ Algoritma Geliştirme ve Programlamaya Giriş – Fahri Vatansever
- ❖ ASP.NET & XML – Yüksel İnan, Nihat Demirli
- ❖ Bilgisayar Öğrenen Herkes İçin – FONO
- ❖ Byte, Chip, PC World, PC Net, PC Magazine, PC Life dergilerinin 1995 – 2005 arası çeşitli sayıları ve ekleri
- ❖ C++ ile Nesneye Dayalı Programlama – Fehmi Demiralp
- ❖ Delphi 2 – Mehmet Çömlekçi
- ❖ Delphi 3 – Marco Cantu
- ❖ Delphi 3 – Serdar Özler
- ❖ Delphi 3 Super Bible – Thurrot, Brent, Tendon, Bagdazian
- ❖ Hatasız Kodlama – Steve Maguire
- ❖ İnternet siteleri (delphiturk.com ve programmersheaven.com gibi)
- ❖ Kim Korkar Unix'ten – Can Uğur Ayfer
- ❖ Linux İşletim Sistemi – Görkem Çetin
- ❖ Macromedia Flash 4.0 – Y.Doç.Dr. Adem Karahoca
- ❖ Microsoft Standart C – Bahattin Bayburan
- ❖ Mikroişlemciler ve Bilgisayarlar – Dr. Haluk Gümüşkaya
- ❖ MS-DOS ile Çalışmak 6.22 – Van Wolverton
- ❖ Office 97 – Faruk Çubukçu
- ❖ PC Donanım – Murat Yılmaz
- ❖ PHP – Dr. Hakkı Öcal PCLife Kasım 2000
- ❖ Üniversiteden kalan ders notlarım ve program örnekleri
- ❖ Yapısal Basic Prensipleri – Halim Korkmaz